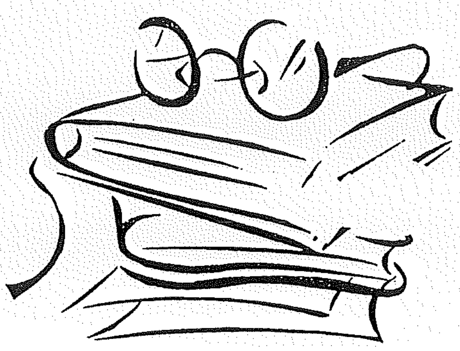
جامعة القاهرة

معهد الدراسات و البحوث الإحصائية

المؤتمر السنوى الرابع والأربعون
للإحصاء وعلوم الحاسب
وبحوث العمليات

مجلد

بحوث العمليات



٧-١٠ ديسمبر ٢٠٠٩

كلمة
المؤتمر السنوى الرابع والأربعون
"للإحصاء وعلوم الحاسب وبحوث العمليات"
٧ - ١٠ ديسمبر ٢٠٠٩

أساتذتى وزملائى الاعـزاء
السيدات والسادة الحضور الكريم

لقد أصبح مؤتمرنا السنوي "للإحصاء وعلوم الحاسب وبحوث العمليات" أحد السمات المميزة لأنشطة معهد الدراسات والبحوث الإحصائية ، كما أصبح محفلاً علمياً رهيناً بما يتضمنه من جلسات علمية متخصصة فضلاً عن الندوات العلمية من مختلف المجالات ذات الصلة ، كما أنه يمثل فرصه سنوية للتواصل بين معهدنا ومختلف الجامعات والمراكز البحثية والوزارات والهيئات والجمعيات والقطاع الخاص ، ويبرز من خلال تلك الفرصة السنوية دور معهد الإحصاء في المساهمة فى المشاركة فى حل القضايا المجتمعية.

أساتذتى وزملائي الأعزاء
السيدات والسادة الحضور الكريم

تعقد فعاليات أنشطة مؤتمرنا السنوي هذا العام تحت شعار:

"نحو إحصاء لتحقيق التنمية الإنسانية"

فقد خلصت خبرات السنوات الماضية عن حقيقة لاتحتاج إلى برهان مفادها ان أى خطه أو برنامج أو مشروع لا يمكن انجازه فى غياب قواعد البيانات وبالتالى الإحصاء كأداة لأتخاذ القرارات. وفى هذا السياق فان اعتزازنا نحو تبنى هذا الشعار. فالتنمية الإنسانية طبقاً لمختلف أبعادها تعتبر وسيلة وغاية لأي جهد تنموي من شتى المجالات الاقتصادية والاجتماعية والثقافية والبيئية. حيث أن التغيرات السريعة والمتلاحقة التى طرأت على العالم خلال العقد الأول من الألفية الثالثة تضيف تحديات جديدة فى سبيل تحقيق التنمية الإنسانية كما أنها تتيح -

فى نفس الوقت فرصاً لتحقيق هذا الهدف . ومن أهم المتغيرات العالميـة التـى طرأت هى الإنتقال من المجتمع الصناعي إلى مجتمع المعرفة.

ولقد أصبح من المسلمات أن المصدر الرئيسى لإنتاج المعرفة يتمثـل فـى التعليم والبحث العلمي والتكنولوجيا ، وإن تطوير التعليم وتشجيع البحـوث هـو السبيل الرئيسي لتحقيق مجتمع المعرفة.

ولا يغيب عنا جميعاً الدور الهام للإحصاء فى حياة البشرية كأداه للمساعدة في إتخاذ القرارات ورسم وتبني السياسات الناجحة على كافة المستويات .

أساتذتى وزملائي الأعزاء

السيدات والسادة الحضور الكريم

إن دور الجامعات ومؤسسات البحث العلمي فى حل مشكلات المجتمع هـو محور رئيسي للمشاركة المجتمعية ، وإن لمعهد الإحصاء بجامعة القـاهرة دوراً واضحاً فى التفاعل مع البيئة المحيطة وخدمة المجتمع وذلك من خلال عدة أنشطة أذكر منها علي سبيل المثال لا الحصر دور المعهد في المشاركة فـى التـسجيل الالكتروني لمكتب التسيق للعام الدراسي الحالى ، وكذلك القوافل التى تم تنظيمها بالتعاون مع مؤسسة النور الخيرية (مغربى).

السيدات والسادة الحضور الكريم

لقد واكب مؤتمرنا فى العام الماضى إدراج معهد الإحصاء بجامعة القاهرة فى مشروع الجودة للتعليم ، حيث خطى المعهد خطوات سوف تساعده فى التأهيـل للتقدم لمشروع CIQAP والذى نري فيه أملاً كبيراً حتى أن يحقق معهدنا رؤيته ورسالته في إقصر وقت ممكن ، وأن ريادة المعهد التى تتمثل فى إنفراده لمنح درجة الدبلوم والماجستير والدكتوراه فى الإحـصاء وعلـوم الحاسـب وبحـوث العمليات تملى علينا أهمية مواكبة العصر من تكنولوجيا المعلومات وطرق التعليم والتعلم والتى تضفي الى علوم الإحصاء وجه جديد بحيث تعمـل علـى تحقيـق مجتمع المعرفة. وفى هذا الإطار فأن المعهد ينظم سلسلة من حلقات العمل ضمن أنشطة وحدة ضمان الجودة لأعضاء هيئة التدريس وسوف تعقد الحلقـة الاولـى يوم الثلاثاء ٢٠٠٩/١٢/٢٢ تحت عنوان :

"صياغة مخرجات التعلم وتحديد أدوات التقييم وتحقيق الترابط
بين المقرر والبرنامج الدراسي والخطة الدراسية"

وسيقوم أ.د.أحمد سيف النصر الأستاذغير المتفرغ بقسم الإحصاء السكاني بالمعهد ونائب رئيس جامعة دبي للشؤون الأكاديمية(سابقا) بالإعداد وأدارة الحلقة.

أساتذتى وزملائي الأعزاء

الحضور الكريم

تتعدد أنشطة مؤتمرنا السنوي هذا العام بحيث يشتمل على (٧) جلسات علمية تحتوي على مايقرب (٥٠) بحثاً في الإحصاء السكاني والتطبيقى والرياضى وعلوم الحاسب والمعلومات وبحوث العمليات.

كما أن معهدنا كعهدة دائماً يتواصل مع الجامعات والمراكز البحثية والوزارات والهيئات والجمعيات من خلال عقد ندوة علمية مركز علمي أذكر منها:

- ندوة : "ماذا بعد مؤتمر القاهرة والسكان (خمسة عشر عاماً)" بالتعاون مع المكتب الاقليمى لصندوق الامم المتحدة للسكان

- ندوة: "دور المشروع العربي لصحة الأسرة بجامعة الدول العربية فى التخطيط وإتخاذ القرارات"
بالتعاون مع المشروع العربي لصحة الأسرة – جامعة الدول العربية.

- ندوة: "رؤية اقتصادية للتنمية الزراعية"
بالتعاون مع مركز البحوث الزراعية – معهد بحوث الأقتصاد الزراعي – وزارة الزراعة

- ندوة: "السكان والتنمية"
بالتعاون مع الاتحاد العربي لحماية البيئة – الامانة العامة

- ندوة "قياس وتحسين الإنتاجية"
بالتعاون مع مصلحة الكفاية الإنتاجية والتدريب – وزارة التجارة والصناعة

- ندوة:

"Statistics for Good Governances: the Ibrahim Index of African Governance" (Dr. Ali Hadi)

بالتعاون مع مركز معلومات ودعم إتخاذ القرار – مجلس الوزراء

ويتواصل المؤتمر بمحاضرتين عامتين من خيرة علماء مصر بالخارج بل سفرائها من خلال مشروع نقل المعرفة والخبرة اكاديمية البحث العلمى والتى نتشرف بهما وهما:

- الاستاذ الدكتور سمير فريد للحديث عن:

"New Studying the Dynamics of International Migration"

- الاستاذ الدكتور عادل المغربي للحديث عن:

"Information Assurance: Challenges Research Directions"

السادة الحضور الكريم:

لايسعنى فى نهاية كلمتى إلا أن أتقدم بخالص الشكر والامتنان لسعادة الأستاذ الدكتور/ حسام محمد كامل رئيس جامعة القاهرة علي كريم رعايته لمؤتمرنا ، والذى لايألوا جهداً في دعم جميع أنشطة المعهد.

كما أتقدم بالشكر والامتنان لجميع الجهات والهيئات الداعمة لمؤتمرنا هذا العام ، وجميع الجهات والهيئات المشاركة فيه ، كما أتقدم بالشكر للجنة المنظمة على الجهد المبذول في تنظيم المؤتمر والشكر موصول لحضراتكم جميعاً علي الحضور والمشاركة . واتمنى أن تخرج جميع أنشطة مؤتمرنا بتوصيات بناءه تساعد متخذى القرار على جميع المستويات.

وأدعو الله العزيز أن يوفقنا جميعاً لما فيه الخير لمصرنا العزيزة.

والسلام عليكم ورحمه الله وبركاته ،،،

عميد المعهد

أ.د. محمد نجيب عبد الفتاح

أمانة المؤتمر والهيئة العلمية

رعاية المؤتمر

تحت رعاية السيد الاستاذ الدكتور/ رئيس الجامعة أ.د.حسام كامل

رئاسة المؤتمر

رئيس المؤتمر أ.د. محمد نجيب عبد الفتاح
وعميد معهد الدراسات والبحوث الإحصائية

أمانة المؤتمر

أمين عام المؤتمر أ.د. يوسف مصطفى محجوب
ووكيل المعهد لشئون خدمة المجتمع وتنمية البيئة

الهيئة العلمية مرتبة أبجدياً

الإســـــم	الجهـــة
ابراهيم السيد البطل	معهد الإحصاء – جامعة القاهرة
ابراهيم فرج عبد الرحمن	كلية الحاسبات – جامعة القاهرة
أبو العلا عطيفي	كلية الحاسبات – جامعة القاهرة
أحمد أمين الشيخ	معهد الإحصاء – جامعة القاهرة
أحمد حسن يوسف	معهد الإحصاء – جامعة القاهرة
أحمد سيف النصر	معهد الإحصاء – جامعة القاهرة
أحمد فـؤاد محمد عطية	رئيس قسم الإحصاء الرياضي بالمعهد
أشرف عبد المنعم الهادى	معهد الإحصاء – جامعة القاهرة
أكرم إبراهيم صلاح	كلية الحاسبات – جامعة القاهرة
الحسينى عبد البر راضى	معهد الإحصاء – جامعة القاهرة
السيد احمـد الشربينى	معهد الإحصاء – جامعة القاهرة
الهام شكري محمـــد	معهد الإحصاء – جامعة القاهرة
أمانى موسى محمد موسى	معهد الإحصاء – جامعة القاهرة
حجازي محمد زاهـر	معهد الإحصاء – جامعة القاهرة

دعد محمد فؤاد	رئيس قسم الإحصاء الحيوي والسكانى بالمعهد
ريم محمد رضا بهجت	عميد كلية الحاسبات والمعلومات – جامعة القاهرة
سحر أحمد نيازى	معهد الإحصاء – جامعة القاهرة
سعد الدين محمد الشيال	معهد الإحصاء – جامعة القاهرة
سمية محمد كامل العطار	قائم بأعمال رئيس قسم الإحصاء التطبيقى بالمعهد
سمية محمود السعدني	معهد الإحصاء – جامعة القاهرة
سمير كامل عاشـــور	معهد الإحصاء – جامعة القاهرة
سناء العلا حنفى احمد	كلية الحاسبات – جامعة القاهرة
سيد مشعال تاج الدين	وكيل معهد الإحصاء – جامعة القاهرة
شعبان عبد الحميد شعبان	معهد الإحصاء – جامعة القاهرة
عاطف محمد عبد المنعم	معهد الإحصاء – جامعة القاهرة
عبد البديع سالم	كلية الحاسبات والمعلومات – جامعة عين شمس
عبد الغني محمد عبد الغني	معهد الإحصاء – جامعة القاهرة
عبد الهادى نبيه أحمد	معهد الإحصاء – جامعة القاهرة
عصام على أمين	معهد الإحصاء – جامعة القاهرة
علاء عبد العزيز احمد	معهد الإحصاء – جامعة القاهرة
على محمد على رجب	معهد الإحصاء – جامعة القاهرة
عماد الدين حسين حسن	جامعة أكتوبر للعلوم الحديثة والآداب
غـزال عبد العزيز عامر	معهد الإحصاء – جامعة القاهرة
فـاروق تمام على شعيب	معهد الإحصاء – جامعة القاهرة
فاطمة عبده إبراهيم الليثي	معهد الإحصاء – جامعة القاهرة
ليلى عبد العزيز على خليل	معهد الإحصاء – جامعة القاهرة
ليلى ناصف عطيـه	معهد الإحصاء – جامعة القاهرة
مجـدى سيـد مصطفى	معهد الإحصاء – جامعة القاهرة
محمد بيومي	كلية الحاسبات – جامعة القاهرة
محمد جمال الدين درويش	عميد كلية الحاسبات – جامعة الاهرام الكندية
محمد حسن جاد الله	رئيس قسم بحوث العمليات بالمعهد
محمد حسن رسمى	كلية الحاسبات – جامعة القاهرة

محمد مصطفى حسن	معهد الإحصاء – جامعة القاهرة
محمـد نجيب عبد الفتاح	عميد معهد الإحصاء – جامعة القاهرة
محمد يحيى البسيونى	معهد الإحصاء – جامعة القاهرة
محمود رياض محمود	معهد الإحصاء – جامعة القاهرة
مرفت حسـن غيـث	معهد الإحصاء – جامعة القاهرة
معتز محمد خورشيد	نائب الجامعة البريطانية بمصر
منى عبد الفتاح خليفة	معهد الإحصاء – جامعة القاهرة
مها مصطفى كمال	معهد الإحصاء – جامعة القاهرة
نجلاء رجاء سعيد حسن	معهد الإحصاء – جامعة القاهرة
نجلاء عبد المنعم مراد	معهد الإحصاء – جامعة القاهرة
هدى أنسي	وكيل كلية الحاسبات – جامعة القاهرة
هشام احمد حفنى	قائم بأعمال رئيس قسم علوم الحاسب بالمعهد
هشام حسن مخلوف	معهد الإحصاء – جامعة القاهرة
هشام محمد عبد السلام	كلية الحاسبات – جامعة القاهرة
هشام نبيه المهدى	كلية الحاسبات – جامعة القاهرة
يوسف مصطفى محجوب	وكيل المعهد لشئون خدمة المجتمع وتنمية البيئة

اللجنة الفنية للمؤتمر

أ. نادية حلمي عواد

سكرتارية المؤتمر

أمين المعهد:

أ. سراج الدين إبراهيم

مساعدو اللجنة الفنية للمؤتمر

أ. جليلة كامل

أ. سوسن عبد الحليم

أ. سالمة محمود أحمد

أ. صفاء محمود

أ. هالة اسماعيل

أ. نسرين إمام

أ. سمر صبري

أ. فوزية أحمد على

أ. أمال عبد الكريم

أ. مصطفى عباس

أ. شيماء إسماعيل شحاته

أ. ريهام محمد عبد الواحد

أ. ريهام أحمد شافعى

العلاقات العامة (التدريب)

أ. مجدي حسن عبد الرحمن

أ. رضا أحمد محمـــد

أ. هناء عبد الحليم

وحدة التصوير والطباعة والتجليد

أ. غريب محمود سالم

الإستعلامات

أ. سلامة عبد الواحد

أ. أبو العلا كمال أبو العلا

[11] Cornuejols G., Sridharan R. and Thizy J.M.. *A Comparison of heuristics and relevations for the capacitated plant location problem.* European Journal and Operational Research, 50:280-297, 1991.

[12] Hajiaghavi M., Mahdian M., and Mirrokni V.. *The facility location problem with general cost function.* Networks, 42(1): 42-47, 2002.

[13] Jain K. and Vazirani V.. *Approximation Algorithms for metric facility Location and K-median problems using the primal- dual schema and Lagrangean relaxation.* Journal of the ACM, 48: 274-296, 2001.

[14] Korupolu M., Plaxton C. and Rajaraman R. , *Analysis of a local search heuristic for facility Location problems.* Journal of Algorithms 37:146-188, 2000.

[15] Lawler E.L., Combinatorial *Optimisation: Networks and Matroids.* Holt, Rinehart, and Winston, New York, 1976.

[16] Levi R., Shmoys D., Swang C.. *LP-based approximation algorithms for capacitated facility location.* In proceedings of the 10th MPS Conference on Integer Programming and Combinatorial Optimisation, pages 206-218, 2004.

[17] Lin J. and Vitter J.. *ε-Approximation with minimum packing constraint violation.* In proceedings of 24th Annual ACM Symposium on Theory of Computing, pages 771-782, 1992.

[18] Lin J. and Vitter J.. *Approximation algorithm for geometric problems.* Inform. Proc. Lett., 44: 245-249, 1992.

[19] Mahdian M., Markakis E. , Saberi A. , and Vazirani V. V.. *A greedy facility Location algorithm analysed using dual fitting.* In Proceedings of the 4th International Workshop on Approximation Algorithms for Combinatorial Optimisation Problems, Springer- Verlag LNCS, 2129: 127-137, 2001.

[20] Mahdian M., Ye Y., Zhang J.. *A 2-Approximation algorithm for the soft-capacitated Facility Location Problem,* RANDOM- APPROX, 129-140, 2003.

[21] Mirchindani P. and Francis R. (editors). *Discrete Location Theory.* John Wiley and sons, Inc. New York 1990.

[22] Raghavan R. and Thompson C.D.. *Randomised rounding.* Combinatorica 7: 365-374, 1987.

[23] Ravi R. , and Sinna A.. *Approximation algorithm for problems combining facility location and network.* Operations Research, 54(1): 73-81, 2006.

[24] Shmoys D., Tardos E., and Aarda K. l. *Approximation Algorithm for Facility Location problem.* Proceedings of 29th ACM Stoc 265-274,1997.

4 Conclusions

In developing an environment for experimenting with different solution approaches we have developed a set of Greedy heuristics to solve the CFLP to provide a feasible integer solution and a lower bound on the optimum solution, which act as a good comparison.

Some of these heuristics are based on column generation to solve the CFLP. We have investigated the trade off between these heuristics in terms of solution quality verses the computational time and the computer memory.
In order to process larger model instances, we have exploited a sparse technology technique in implementing our algorithm. Thus we have developed a sparse data structure that we exploit in our algorithm implementation.

The computer environment is driven by a specification file whereby any combination of our greedy heuristics can be exploited in the solution approach by turning on or off the appropriate settings, thus the computer implementation provides a flexible environment for future researchers to experiment with different strategies for solving the CFLP

References

[1] Aardal K. , *Capacitated facility location: Separation algorithms and computational experience,* Mathematical Programming, 81:149-175, 1998.

[2] Aardal k., *Reformulation of capacitated facility location problems: How redundant information can help.* Annals of Operations Research, 82(0): 289-308, 1998.

[3] . Barahona F and Anbil R.. *The volume algorithm: Producing primal solutions with a subgradient method.* Mathematical Programming, 87:385-399, 2000.

[4] Barahona F. and Chudak F.A., *Near –optimal solutions to large scale facility location problems RC 21606*, and Yorktown Heights, N.Y: IBMT J. Watson Research Centre, 1999.

[5] Beasley J.E.. *An algorithm for solving large capacitated warehouse location problems.* European Journal of Operational Research, 33: 314-325, 1988.

[6] Beasley J.E.. *OR-Library: Distributing test problems by electronic mail,* Journal of the Operational Research Society, 41: 1069-1072, 1990.

[7] Beasley J.E.. *Lagrangean heuristics for location problems.* European Journal of Operational Research, 65: 383-399, 1993.

[8] Chudak F., and Shmoys D. B.. *Improved approximation algorithms for capacitated facility location.* Unpublished Manuscript, 1998.

[9] Chudak F. A., Shmoys D. B.. *Improved approximation algorithm for a capacitated facility location problem.* Lecture notes in computer science, 1610: 99- , 1999.

[10] Chudak F. A. and Williamson D.P.. *Improved approximation algorithms for Capacitated facility Location problems.* Unpublished Manuscript. 1998 and preliminary version appeared in IPCO,1999.

Table 3 below is comparison between the results of Beasley's problem size 25×50 and the results of our Greedy heuristics. The problem size is 25×50, has three data files for the same problem; the difference is being in the fixed costs. We obtained a lower bound LB together with a first upper bound UB on the optimum solution.

Comparing the results, we were able to get in most of the cases the same or a very close UB to Beasley's. This shows that our Greedy Heuristics provide reasonable results for the comparative purposes, which proves that our greedy heuristics works well, and that was the main reason behind this comparison. Also the cray-1 system still has more capability than PC's even at the time when it was invented.

Also comparing the computational time using a computer environment, we were able to get in all the cases, a much shorter running time than Beasley's. Recall that Beasley ran his problems on Cray-1S using the CFT compiler.

Notice that Cray-1S is a **supercomputer** that led the world in terms of processing capacity, particularly speed of calculation. The CFT compiler is available for the CRAY series of computer systems, the CRAY-2 computer system, and for CRAY-1 computer systems, and it operates under both COS and UNICOS, the Cray operating systems.
CFT takes full advantage of the unique hardware architecture of Cray computer systems and by doing so greatly enhances their performance. Thus,users benefit from hardware and software that work together to achieve maximum performance.

Beasley's data files	Sol.	Beasley's sol.	Method 1	Method 2
Cap 81	**LB**	-	845751.000	845751.000
	UB	838499.288	845889.562	845889.562
	Time	3.8	0.780	0.740
	Error	-	0.016	0.016
Cap 82	**LB**	-	910751.369	910751.369
	UB	910889.563	911277.312	910889.563
	Time	1.8	0.740	0.710
	Error	-	0.058	0.015
Cap 83	**LB**	-	975567.036	975567.036
	UB	975889.563	975889.563	975889.563
	Time	1.8	0.750	0.750
	Error	-	0.033	0.033
Cap 84	**LB**	-	1067469.475	1067469.475
	UB	1069369.525	1090977.450	1069369.525
	Time	4.0	0.790	0.800
	Error	-	2.155	0.178

Table 3: A comparison between the results of Beasley's problem size 25×50 and the results of our Greedy heuristics.

3.2 Results

We present the computational results and the analysis for our Greedy heuristics that were explained in section 2.

In **Table 2** below, we report a summary of the problem results for the Greedy Heuristics. This table contains the results for problems with sizes between 100-300. These problem sets being randomly generated as detailed above.

The table is divided into two cases; the first case is the results of the Greedy Heuristics runs with each Method. The second case is the results using the same representation of the CFLP while also applying the **"cut off point"** technique explained in subsection 2.2.
This table is a comparison between the two cases, to show the successful applications of this technique. By comparing the running time in both cases, we notice a big difference, while at the same time there is no detriment in the solution quality .The promising results of this investigation is that the cut off point heuristic is much more effective in the larger sized problems, generally finding the same lower bound (LB) but obtaining a superior upper bound (UB). The error terms correspond to the percentage value of
The formula;

$$\frac{UB-LB}{UB}.$$

We were able to solve problems up to 300×300. **Notice** that in all the results of our Greedy Heuristics, the LB is the solution of the LP-relaxation problem, while the UB is the ***first feasible integer solution*** of the P_{int} problem.

We are interested in obtaining a first integer solution because it is very expensive in computational time if we look for more than one feasible integer solution. This is the reason why in some of our Greedy heuristics results the quality of the UB was not that good.

Problem Size i×j	Sol.	WITOUT CUT OFF POINT		WITH CUT OFF POINT	
		Method 1	**Method 2**	**Method 1**	**Method 2**
100x100	LB	11089.600	11089.600	11104.436	11104.436
	UB	12860.036	11825.640	12883.839	11859.539
	Time	26.07	61.99	14.78	30.56
	Error %	13.80	6.20	13.80	6.36
200x200	LB	19542.300	19542.300	19545.50	19545.5
	UB	22054.726	19826.729	21043.52	19792.521
	Time	808.80	725.49	484.66	218.40
	Error %	11.40	1.40	7.12	1.25
300x300	LB	28351.800	28351.800	28351.800	28351.800
	UB	30242.766	29125.771	30262.135	28992.636
	Time	5596.96	5507.79	1215.79	4328.10
	Error %	6.25	2.70	6.30	2.21

Table 2: A comparison between the results of the Greedy heuristics with and without the "cut off point" technique.

combinations of (i, j). Obviously, too low a percentile point leads to a poor solution, but a very quick computational time, while too high a value results in little improvement in computational time. We report some successful applications of this technique in section 3.

3 Our Results

All the Greedy heuristics were implemented in FORTRAN, with FortMP as a solver when appropriate. We test the performance of our Greedy heuristics on different problems by using alternative benchmark models (test problems).

3.1 The Test Problems

The selected model collection was drawn from:

1- J.E. Beasley [5], we selected one instance of the problem of size $i \times j = 25 \times 50$. This data was obtained from his OR-Library [6]. Problems VIII are 25×50. Beasley used a Lagrangean Relaxation algorithm of a mixed-integer formulation of the CFLP, together with problem reduction tests derived from both the original problem and the Lagrangean Relaxation. He was able to solve instances of up to 300 potential locations and 300 customers. He programmed his problems in FORTRAN and ran them on a Cray-1S using the CFT compiler with maximum optimisation. **Notice** that his solutions are the optimum integer solutions for the CFLP.

3.2 Data Generation

We use instances generated as in the work of Barahona and Anbil [3], and the work of Cornuejols, Sridharan and Thizy [11], which are as follows:

- We generate the demand points j and the facility points i uniformly at random in a $[0,1] \times [0,1]$ gnd.
- The demand values d_j are generated from a uniform distribution with mean 35 and standard deviation 5, U [5,35].
- The capacities S_i are generated from a U [10,160] distribution, and are then rescaled so that $\sum_{i \in F} S_i / \sum_{j \in D} d_j$ was set equal to the parameter factor; this was either set at 10, and in some cases 1.5.
- The fixed cost f_i of each site was set using the formula $f_i = U[0,90] + U[100,110]\sqrt{S_i}$.
- The unit transportation costs c_{ij} correspond to the Euclidean distance scaled by 10.

This constraint tightens the lower bound representation of the CFLP. The value of p is determined as follows; we sort all of the facilities in descending order of capacity. Then p is such that in the sorted order of facilities

$$\sum_{i=1}^{p-1} S_i < \sum_{j=1}^{n} d_j \leq \sum_{i=1}^{p} S_i \,. \tag{8}$$

Although the right hand side of the constraint (7) need not be bounded, as a consequence of our experiments using $p+2$ speeds up considerably the computational time with no detriment to the solution quality.

2.1 The Greedy heuristics (Methods)

The following table (Table 1) gives a summary of the greedy heuristics (Methods) applied in solving different problems.

Methods	Analysis of the (LP) solution	Modification of (IP) model to set the optimum solution
Method 1	Let (x^*, y^*) be the optimal solution of the LP problem and $F' \subseteq F, D' \subseteq D$, such that, $x_{ij}^* = 0$, $\forall i \in F'$, $j \in D'$ and $y_i^* = 0$, $\forall i \in F'$.	- Set $y_i = 0$, $\forall i \in F'$ - Set $x_{ij} = 0$, $\forall i \in F'$, $j \in D'$ - Rows $x_{ij} \leq y_i$ are relaxed $\forall i \in F'$, $j \in D'$.
Method 2	Let (x^*, y^*) be the optimum solution of the LP problem and $F' \subseteq F$, such that $y_i^* = 1$, $\forall i \in F'$.	- Set $y_i = 1$, $\forall i \in F'$. - Rows $x_{ij} \leq y_i$ are relaxed $\forall i \in F'$, $j \in D$.

Table 1: describes how the greedy heuristics (Methods) work

We now present a short rational for each heuristic. In **Method 1**, we view any x or y that takes the value 0 in the relaxed LP solution, is likely to be 0 in the integer solution. Consequently we set these x's and y's to zero in the IP problem representation, and we relax the corresponding constraints of (4). **Method 2** has a different view; we keep all the facilities opened in the IP problem, for each $i \in F'$ where $y_i = 1$ in the solution of the LP-relaxation problem plus we relax the corresponding constraints of (4).

2.2 The Cut off Point

Also we describe **a techniques** that we used to improve the running time., we have analysed the distribution of the distance matrix c_{ij}. Sorting this by value, we have looked at setting different percentile points of this distribution as **cut off points** to rule out various

In addition, Jain and Vazirani [13] have extended their work on the UCFLP to consider the soft CFLP. Their idea to approach the capacitated case, is to move the capacity constraints to the objective function using Lagrangean multipliers, they used also a primal dual algorithm, and were able to achieve an approximation guarantee of 4 for the capacitated case in which multiple copies of facilities can be opened. Levi, Shmoys and Swamy [16] presented an algorithm that is based on integer rounding of the optimal fractional solution to the LP-relaxation. They used this solution to guide the decomposition of the input into a collection of single-demand-node capacitated facility location problems, which are then solved independently. In the case where all facility-opening costs are equal, they show that their algorithm is a 5-approximation guarantee factor. Their algorithm relied on a decomposition of the input into instances of the single-demand capacitated facility location problem.

However, Barahona and Chudak [3,4] used a new heuristic based on the Lagrangean Decomposition and the volume algorithm of Barahona and Anbil [4], to approximately solve a linear programming relaxation and randomised rounding of Raghavan and Thompson [22] to find feasible integer solutions.

A large number of approximation algorithms with different techniques have been proposed recently for various applications of the CFLP, see for example [1,2,5,7,11,12,19,23].

Notice that to measure the quality of the solution found, researchers have introduced a constant approximation guarantee factor (C_{OPT} / C_{LP}), where C_{OPT} is the optimum value for the (P_{int}) formulation, and C_{LP} is the optimum value for the (P), as it was explained above.

2 A Set of Greedy Methods to Solve CFLP

In this paper we focus on one of the hardest models of the facility location theory: the capacitated facility location problem (CFLP). We consider the variety of the CFLP, where the client $j \in D$ maybe serviced by several facilities. We introduce some greedy heuristics; we will present them by **Methods**. All of these methods are based on analysing the solution to the LP-relaxation representation for the CFLP.

In carrying out our experimental investigations for large instances of the problems, it is impractical to have all the linking constraints,

$$x_{ij} \le y_i, \qquad \forall i \in F, j \in D. \tag{3}$$

This drastically weakens the relaxed LP lower bound (LB). We introduce a new constraint that helps in improving this value.

Adding New Constraints:

Given the LP-relaxation of the CFLP in subsection 1.1, we add a new constraint with regards to the minimum desired number p of facilities that need to be open for the problem,

(P) Minimise $\qquad \sum_{j \in D} \sum_{i \in F} d_j c_{ij} x_{ij} + \sum_{i \in F} f_i y_i$

Subject to $\qquad$ (1), (2), (3) and (5)

$$y_i \leq 1, \qquad \forall i \in F \qquad (6)$$

Notice that in the LP -relaxation of (P_{int}) above, we replace constraint (4) by (6).

1.2 Previous and Related Work

There are two variants of the capacitated problem, depending on whether each location's demand must be assigned to only one facility, or the demand may be fractionally split among more than one (completely) open facility. For a review of this problem, see the textbook of Lawler [15]).

Solving the capacitated facility location problem, there is no such heuristic that always works well in practice. Part of the reason is that the linear programming relaxation is known not to be tight.

The CFLP and variations of it was extensively treated in the literature (see for example the book of Mirchandani and Francis [21]), and for more review work see the work of Shmoys, Tardos and Aardal [24], they gave a 5.69 approximation guarantee factor for their algorithm for solving the variant of the CFLP in which we split a customer's demand among several facilities, and 7.62 approximation factor for the variant in which each customer's demand may be served by exactly one facility. Their algorithm is based on a randomised variant of the filtering technique of Lin&Vitter [18, 17].

Also, Chudak and Shmoys [8,9] improved the approximation factor of Shmoys, Tardos and Aardal to 3. Their algorithm are based on starting with the optimal solution to a linear programming relaxation (*P*) of the original problem, and then rounding this fractional solution using their randomised rounding with a back up technique to a feasible integer solution. Moreover, Korupolu, Plaxton and Rajaraman [14] extended their local search algorithm for solving the metric UCFLP explained in section 2.2, to solve the CFLP version. Using this simple local search heuristic they were able to produce a solution with an approximation factor of $8+\varepsilon$, where $\varepsilon > 0$.
Moreover, Mahdian, Ye and Zhang [20] consider the variant of the CFLP in which one can open multiple copies of any facility (soft CFLP). They were able to improve the approximation factor of Shmoys, Tardos and Aardal [24]. Their algorithm is based on an improved analysis of an algorithm for linear facility location problem, and use a " bifactor " approximate-reduction from this problem to the soft CFLP (For more on the concept of the bifactor see [20]).

Also, Chudak and Williamson [10] solve the same problem of Mahdian, Ye and Zhang above using ideas from the analysis of the local search heuristic. They were able to simplify and improve Korupolu, Plaxton and Rajaraman [14] analysis of the heuristic, showing an improvement in the approximation factor, achieving $(6+\varepsilon)$.

1.1 The Description of the FLP

The facility location problem (FLP) seeks to locate a number of facilities to serve a number of customers, thus there is a set of potential facility locations F; opening a facility at location $i \in F$ has an associated nonnegative fixed cost f_i, and has either a limited or unlimited capacity S_i of available supply. There is set of customers (clients) or demand points D that require service; customer $j \in D$ has a demand d_j that must be satisfied by the open facilities. If a facility at location $i \in F$ is used to satisfy part of the demand of client $j \in D$, then there is a service or transportation cost incurred, which is often proportional to the distance from i to j, denoted by c_{ij}.
The goal is to determine a subset of the set of potential facility locations at which to open facilities, and to assign the clients to these facilities without violating the capacity constraints, so as to minimise the overall total cost, that is; the fixed cost of opening the facility plus the total service cost.

The algebraic formulation of the problem which is a Mixed-Integer programs, we call it (P_{int}), is as follows:

(P_{int}) Minimise $\displaystyle\sum_{j\in D}\sum_{i\in F} d_j c_{ij} x_{ij} + \sum_{i\in F} f_i y_i$

Subject to

$$\sum_{i\in F} x_{ij} = 1, \qquad \forall j \in D, \qquad (1)$$

$$\sum_{j\in D} d_j x_{ij} \le S_i y_i, \qquad \forall i \in F, \qquad (2)$$

$$x_{ij} \le y_i, \qquad \forall i \in F, j \in D, \qquad (3)$$

$$y_i \in \{0,1\}, \qquad \forall i \in F, \qquad (4)$$

$$x_{ij} \ge 0, \qquad \forall i \in F, j \in D, \qquad (5)$$

Notice that if we are solving an uncapacitated case then we can remove inequalities (2).

The variables x_{ij} represent the fraction of the demand of customer $j \in D$ that is serviced by the facility opened at location $i \in F$, while y_i are binary variables, (y_i=1) indicates that a facility at location $i \in F$ is open.
The objective function represents the total cost, that is; the fixed cost and the transportation costs. Inequalities (1) ensure that each customer's demand must be met. The inequalities (2) are capacity constraints ensuring that each facility is operated within its capacity. While constraint group (3) ensure that if any part of customer j's demand is met by site i, then this facility must be open. Although these are redundant, they provide tighter constraints for the relaxed linear programming representation of the problem.

In all the approximation algorithms based on linear programming, to solve a Facility Location Problem (FLP), first we need to consider LP-relaxation of (P_{int}), which is simply as follows:

(P) Minimise $\displaystyle\sum_{j\in D}\sum_{i\in F} d_j c_{ij} x_{ij} + \sum_{i\in F} f_i y_i$

Greedy Heuristics to Solve the Capacitated Facility Location Problem

Eiman Jadaan Alenezy

Abstract

The facility location problem has been the focus of much research in the Operational Research literature. We investigate solution approaches for large instances of the capacitated facility location problem (CFLP). We introduce a new tightening constraint and dual estimates that lead to improved lower bounds (LB). In our computer environment, we describe a number of Greedy heuristics that provide good upper bound solutions with a lower bound from a comparative point of view.

We show that for small instances of the CFLP, it is effective to have very tight linking constraints. We report results that illustrate the trade off between these constraints, solution time, and solution quality. We detail some techniques that improve the computational running time with no loss in solution quality.

Also we show through solving larger instances of the CFLP that our algorithm scales up well in terms of both computational time and solution quality. Moreover, we develop a sparse technology technique in implementing our algorithm and a flexible computer environment.

Keywords and phrases:
FLP, CFLP, UCFLP, Greedy heuristics, cut off point

1 Introduction

The facility location problem is a long established research area in Operational Research.
Clearly there are many applications in real life, such as in production planning, hospitals, networks, ambulance stations, and many other practical applications.
In this paper we are concerned with the mathematical approaches to solve large instances of this problem, rather than investigating its applications.

We will focus on one variant of the problem: the capacitated facility location problem (CFLP). In many formulations of the CFLP, it is assumed that each demand point can be supplied by only one open facility, which is the simplest case of the problem. We consider the case where each demand point can be supplied by more than one open facility.

We develop a number of Greedy heuristics that provide good solutions for known problems in the literature.

---

* Dr Eiman Jadaan Alenezy, Math Department, Public Authority for Applied Education and Training. Kuwait.

the FAHP-TOPSIS is farther from the negative ideal solution (CC_1=0.8006) than in the AHP-TOPSIS (CC_1= 0.7417). Also the difference between the best ranked alternative and the other less ranked alternatives is more recognized in the FAHP-TOPSIS as shown in Table 10. The difference between the relative closeness to the negative ideal solution of best ranked alternative A1 (0.8006) and the worst ranked alternative A3 (0.1081) in the FAHP-TOPSIS is greater than the relative closeness to the negative ideal solution of best ranked alternative A1 (0.7439) and the worst ranked alternative A3 (0.1933) in the FAHP-TOPSIS.

Table 10: AHP-TOPSIS and FAHP-TOPSIS alternatives ranking

Alternatives	AHP-TOPSIS Relative closeness to the negative ideal solution CC_i^*	Rank	FAHP-TOPSIS Relative closeness to the negative ideal solution CC_i^*	Rank
A1	0.7439	1	0.8006	1
A2	0.3989	2	0.4627	2
A3	0.1933	3	0.1081	3

Then using the FAHP affects the efficiency of representing the alternatives ranking. It makes the best solution farther from the negative ideal solution and the worst solution closer to the negative ideal solution. It will be easier for the decision maker to discriminate between the ranked alternatives especially when the number of alternatives under consideration is increased. That is how the fuzzy based proposed approach gathers the simplicity representation of the hierarchical location selection problem while considering the data uncertainty. This adds robustness value to decision making approach providing the ability to map the decision maker's linguistic perferences to fuzzy numbers.

6. The Importance of Weighting Phase

Solving the same adopted case study while neglecting the weighting phase affects the value of the relative closeness to the negative ideal solution (CC_i^*), though not affecting the alternatives rank in this case. When ignoring the criteria weights they are assumed to be equally important ($w_i=1/n$ for i = 1, 2,..., n). The results are presented in Table 11 and compared with the weighted criteria results from AHP-TOPSIS and FAHP-TOPSIS. The rank of the alternatives in decreasing order is A1, A2, and A3, which is the same in the weighted AHP, weighted FAHP, and unweighted cases. But the relative closeness to the negative ideal solution is obvious in the two weighted cases (AHP-TOPSIS and FAHP-TOPSIS) than in the unweighted case.

Table 11: weighted and unweighted rankings

	Weighted AHP		Weighted FAHP		Unweighted	
	CC_i	Rank	CC_i	Rank	CC_i	Rank
A1	0.7439	A1	0.8006	A1	0.6704	A1
A2	0.3989	A2	0.4627	A2	0.4761	A2
A3	0.1933	A3	0.1081	A3	0.2164	A3

b. Determine location ranks using TOPSIS (Table 9).

i. Determine the ideal solution A+ and negative ideal solution A^- (Table 8).

$$A_j^+ = \max v_{ij} \quad i = 1,2,...m$$

$$A_j^- = \min v_{ij} \quad i = 1,2,...m$$

Table 8: FAHP-TOPSIS Ideal and negative ideal solutions

	C1	C2	C3	C4	C5	C6
A1	0.0390	0.0377	0.1548	0.0615	0.0502	0.0368
A2	0.0369	0.0472	0.0817	0.0495	0.0783	0.0505
A3	0.0341	0.0451	0.0295	0.0289	0.0515	0.0527
A^+	0.0390	0.0377	0.1548	0.0615	0.0783	0.0505
A^-	0.0341	0.0472	0.0295	0.0289	0.0502	0.0368

ii. Calculate the separation measure

$$D_i^+ = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - A_j^+\right)^2} \quad i = 1,2,...,m$$

$$D_i^- = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - A_j^-\right)^2} \quad i = 1,2,...,m$$

iii. Calculate the relative closeness to the negative- ideal solution CCi^*

$$CC_i^* = \frac{D_i^-}{D_i^+ + D_i^-} \quad i = 1,2,...,m$$

iv. Rank the preference order

The alternative with highest distance from the negative ideal solution (CC_i close to 1) is closest to the ideal solution and can be chosen as the best solution. The alternative with the nearest distance from the negative ideal solution (CC_i close to 0) has the highest distance from the ideal solution and is taken as the worst solution.

Table 9: The relative closeness to the negative- ideal solution of the three alternatives

Alternatives	D_i^+	D_i^-	Relative closeness to the negative ideal solution CC_i^*	Rank
A1	0.0323	0.1299	0.8006	1
A2	0.0748	0.0644	0.4627	2
A3	0.1325	0.0161	0.1081	3

c. Select the optimal location.

It is apparent that A1 is the best alternative as it has the larger distance far from the negative ideal solution. This value indicates a better performance of that alternative than the other lower ranked alternatives. The alternatives ranking doesn't differ in the AHP-TOPSIS than in the FAHP-TOPSIS. No longer, the best alternative A1 in

In comparison with the traditional analytical hierarchical method, considering the fuzzy element in the proposed FAHP-TOPSIS model shows superiority in modelling the uncertainty and vagueness associated with the pairwise comparison of different criteria. This model is also less demanding towards the decision makers. Fuzzy based models are more robust decision making approach because of its ability to map the decision maker's perception to triangular fuzzy numbers in place of exact numbers. While Chan [1] developed a Fuzzy AHP model adopting the same case study of Min [10], these fuzzy criteria weights are combined with TOPSIS method to compose a hybrid FAHP-TOPSIS model that can be applied in location selection decision. The proposed fuzzy based model is developed through three stages which are alike to the flowchart of the previous discussed AHP-TOPSIS proposed model (Figure 3), except that the weights created in stage 2 which are used in criteria weighted evaluation matrix of TOPSIS are triangular fuzzy numbers from FAHP. These stages are simply viewed by the framework in Figure 4.

Stage 1:
In this stage the location alternatives are determined and the decision hierarchy is formed (Figure 1). The main objective is presented in the first level, criteria in the second level, and alternatives in the third level. When the decision hierarchy is approved by the decision makers the next stage is considered.

Stage 2:
Use FAHP to assign weights for the criteria that will be used in the location selection decision. When Chan [1] applied FAHP to the location selection problem of Min [10] the weights of the criteria are determined (w_j).

$$W = [0.11, 0.13, 0.30, 0.14, 0.18, 0.14]$$

If the weights are ordered according to the degree of importance, it is apparent that the FAHP weights differ from the AHP weights. The degree of importance of C5 and C2 is changed. In AHP, C2 is the second important criteria, while in FAHP it is the fifth important weight. C5 is the sixth important weight in AHP, while it is the second important weight in FAHP. Proceeding to the next FAHP-TOPSIS stages, it can be shown that these changes in the importance of weights don't affect the alternatives ranking.

Stage 3:

a. Evaluate location alternatives
   i. Available alternatives are compared under each of the criteria separately in TOPSIS to establish the evaluation matrix A.
   ii. Obtain weighted evaluation matrix V using the fuzzy weights of the FAHP (W), and the previously determined evaluation matrix A ($V= W*A$).

Table 7: FAHP-TOPSIS weighted evaluation matrix

	C1	C2	C3	C4	C5	C6
A1	0.0390	0.0377	0.1548	0.0615	0.0502	0.0368
A2	0.0369	0.0472	0.0817	0.0495	0.0783	0.0505
A3	0.0341	0.0451	0.0295	0.0289	0.0515	0.0527

alternatives counts as an advantage in addition to the previously discussed AHP-TOPSIS advantages.

Table 6: AHP and AHP-TOPSIS alternatives ranking

Alternatives	AHP priority	Rank	Relative closeness to the negative ideal solution CC_i^*	Rank
A1	0.374	1	0.7439	1
A2	0.346	2	0.3989	2
A3	0.281	3	0.1933	3

5. A Fuzzy AHP-TOPSIS Proposed Model

Though AHP-TOPSIS has the simplicity in representing the problem in a hierarchical form, and the clarity in discriminating the best alternative, it can not deal with the uncertainty or the vagueness of the data. Some of the criteria in the location selection problem are subjective and qualitative in nature, which make it very difficult for the decision maker to express preferences using exact numerical values. It would also be difficult to provide exact pairwise comparison judgements. Using a crisp model like AHP or AHP-TOPSIS will force the decision maker to ignore some details when solving the location decision problem. So, it is more desirable for decision makers to use interval or fuzzy evaluations to handle the vagueness of the data involved in such multi criteria decision making problems. AHP is recognized as one of the most popular multi criterion decision making techniques. But this method is often criticized because of its insufficiency to consider the uncertainty associated with the expressed linguistic criteria and its inappropriateness for crisp ratio representation.

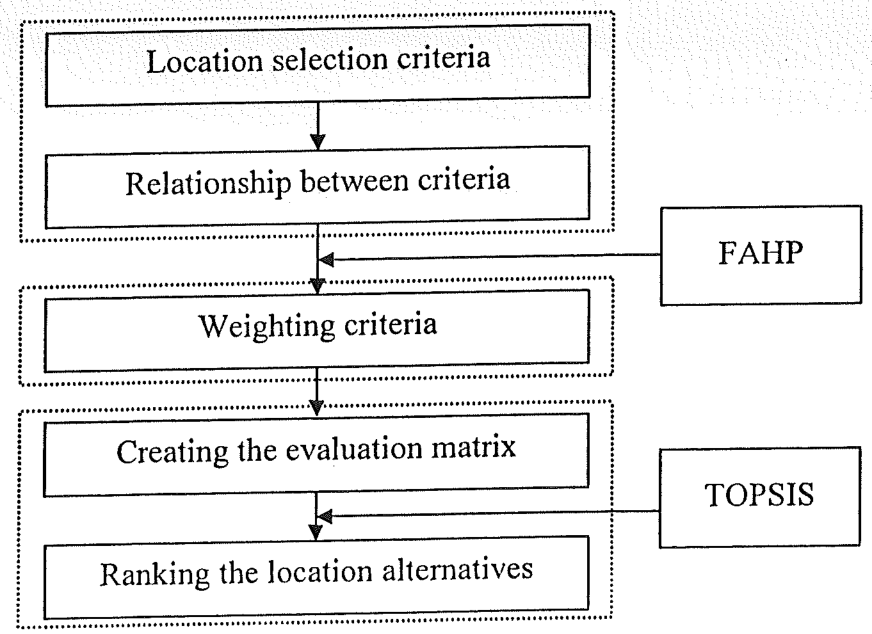


Figure 4: Framework of the proposed FAHP-TOPSIS model

Table 4: ideal and negative ideal solutions

	C1	C2	C3	C4	C5	C6
A1	0.0539	0.0526	0.1115	0.0721	0.0340	0.0431
A2	0.0510	0.0657	0.0588	0.0580	0.0531	0.0591
A3	0.0471	0.0627	0.0456	0.0339	0.0348	0.0618
A^+	0.0539	0.0526	0.1115	0.0721	0.0531	0.0618
A^-	0.0471	0.0657	0.0456	0.0339	0.030	0.0431

ii. Calculate the separation measure

$$D_i^+ = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - A_j^+\right)^2} \quad i = 1,2,\ldots,m$$

$$D_i^- = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - A_j^-\right)^2} \quad i = 1,2,\ldots,m$$

iii. Calculate the relative closeness to the negative- ideal solution CCi^*

$$CC_i^* = \frac{D_i^-}{D_i^+ + D_i^-} \quad i = 1,2,\ldots,m$$

iv. Rank the preference order (Table 5).
The alternative with highest distance from the negative ideal solution (CC_i close to 1) is closest to the ideal solution and can be chosen as the best solution. The alternative with the nearest distance from the negative ideal solution (CC_i close to 0) has the highest distance from the ideal solution and is taken as the worst solution.

Table 5: The relative closeness to the negative- ideal solution of the three alternatives

Alternatives	D_i^+	D_i^-	Relative closeness to the negative ideal solution CC_i^*	Rank
A1	0.0267	0.0776	0.7439	1
A2	0.0562	0.0373	0.3989	2
A3	0.0793	0.0190	0.1933	3

c. Select the optimal location.
It is apparent that A1 is the best alternative as it has the larger index value that indicate a better performance of that alternative.

The alternatives ranking in decreasing order is A1, A2, and then A3. This rank is the same as the AHP alternatives ranking, but it is obvious from Table 6 that the difference between the best location alternative and the other less preferred alternatives is more contrast than that difference among the AHP alternative priority values. In brief, the AHP-TOPSIS model makes the best location alternative easily discriminated. This ability of making the best alternative more recognized than the other lower ranked

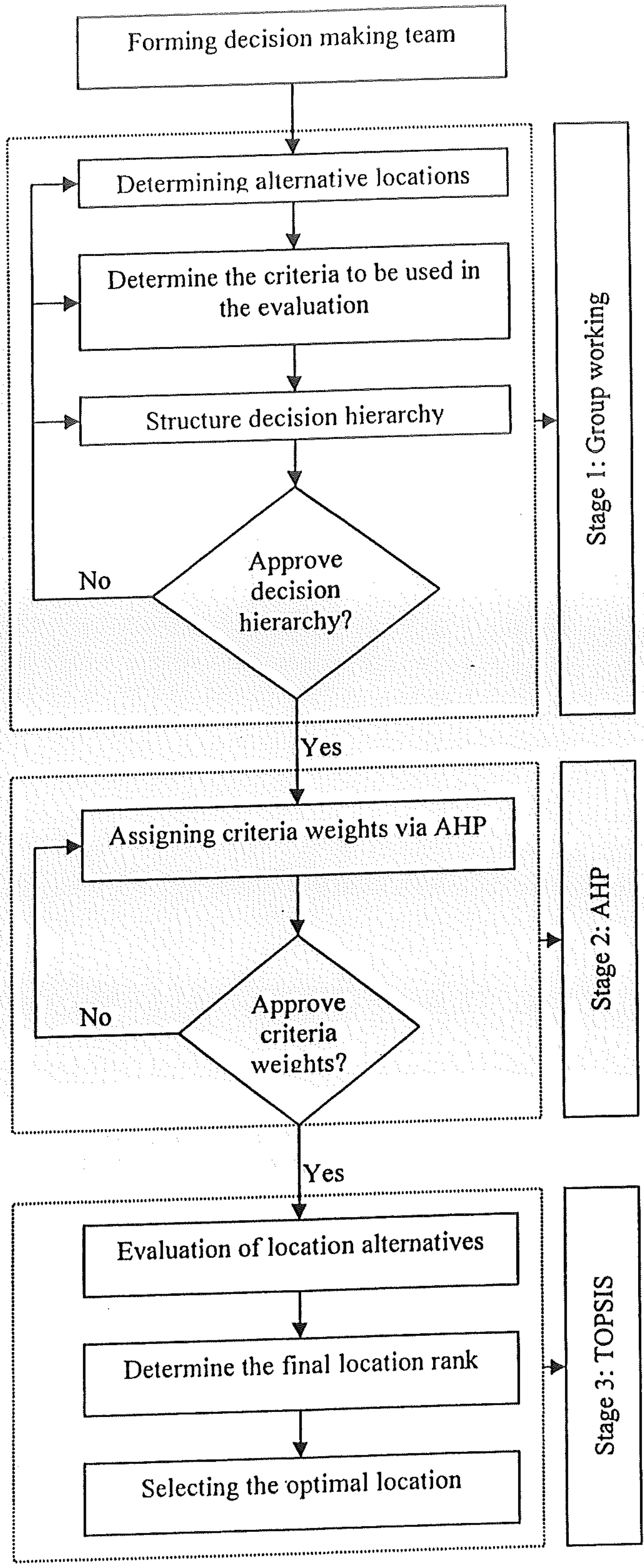


Figure 3: Flowchart of the AHP-TOPSIS proposed model for site location selection

The proposed AHP-TOPSIS model consists of three basic stages represented in Figure 3. Stage1: Identifying the model objective, criteria, and alternatives, stage2: Apply AHP to assign weights to the specified criteria, and stage3: Evaluating available alternatives and determining the final ranking using TOPSIS. Using the adapted case study of Min [10], which was applied previously using AHP, the proposed model of AHP-TOPSIS can be applied in the follows stages:

Stage 1:

a. Alternative locations are determined.
b. Forming the decision hierarchy (Figure 1), objective is in the first level, criteria are in the second level, and alternatives are in the third level.
c. Taking the decision of approving the decision hierarchy by the decision-making team.

Stage 2:

a. Use AHP to assign weights for the criteria used in the location selection.
b. Form the pairwise comparison to determine the criteria weights (w_j).

$$W= [0.152, 0.181, 0.216, 0.164, 0.122, 0.164]$$

Stage 3:

a. Evaluate location alternatives
   i. The weights of each criterion with respect to alternatives are determined and represented in an evaluation matrix (Table 2).

Table 2: the evaluation matrix

	C1	C2	C3	C4	C5	C6
A1	0.3546	0.2905	0.5162	0.4395	0.2788	0.2629
A2	0.3354	0.3629	0.2724	0.3538	0.4352	0.3605
A3	0.3100	0.3466	0.2113	0.2067	0.2860	0.3766

   ii. Obtain weighted evaluation matrix V using the weights of the AHP (W), and the previously determined evaluation matrix (Table 3).

Table 3: the weighted evaluation matrix

	C1	C2	C3	C4	C5	C6
A1	0.0539	0.0549	0.1084	0.0721	0.0340	0.0431
A2	0.0509	0.0686	0.0572	0.0580	0.0531	0.0591
A3	0.0471	0.0655	0.0443	0.0339	0.0348	0.0618

b. Determine location ranks using TOPSIS
   i. Determine the ideal solution A+ and negative ideal solution A^- (Table 4).

$$A_j^+ = \max v_{ij} \quad i = 1,2,\ldots m$$

$$A_j^- = \min v_{ij} \quad i = 1,2,\ldots m$$

performed in the AHP is relatively large and consumes much of time and computational effort.

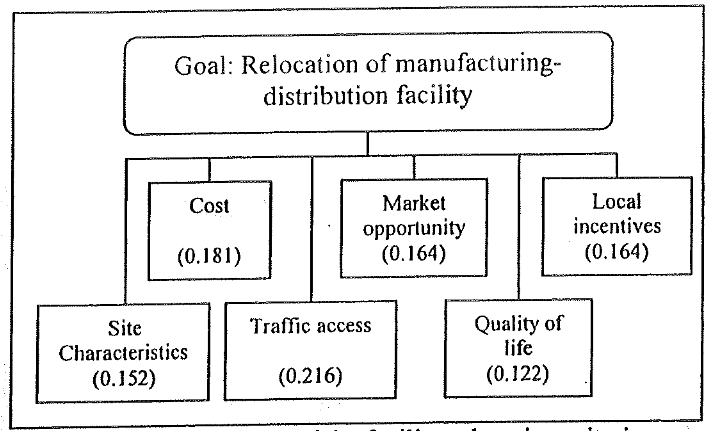


Figure 2: Weights of the facility relocation criteria

4. An AHP-TOPSIS Proposed Model

AHP model is a robust and flexible model. It can represent the problem in a simple hierarchical form, representing the mail goal of the problem at the top level, the criteria affecting the decision in the lower level, followed by the available alternatives in the lower level of the hierarchy tree. On the other hand it lacks the advantage of coming to a solution in a few steps. It requires a large number of pairwise comparisons. That is why the full AHP solution is only practically usable if the number of criteria and alternatives is sufficiently low. The TOPSIS approach also has some merits as: 1) The rationality and ability of understanding of its logic, 2) The computations are straightforward, 3) Searching of best alternatives for each criterion is depicted in a simple mathematical form,) The importance weights are incorporated into the comparison procedures. The newly developed model combines the main advantages of both the AHP and TOPSIS models into one model. These advantages make the proposed AHP-TOPSIS model more potential, as it avoids a serious computational difficulty and achieve the alternatives ranking in a simple mathematical form.

Making use of the advantages of both the AHP and TOPSIS models, the AHP is used to analyze the structure of the problem and to determine the criteria weights, while the TOPSIS is used to develop the evaluation matrix simultaneously. That is how the AHP-TOPSIS model is applied to obtain the final ranking of the determined alternatives. This approach avoids an unreasonably large number of pairwise comparisons if full AHP decision approach is used. For example, if there are n criteria which have been assigned the importance weights and m alternatives, then to run a full-AHP solution there are $n.m.(m-1)/2$ pairwise comparison remaining to be performed [11]. This weak point in the full AHP approach is counted as an advantage in the AHP-TOPSIS approach.

such as aluminium bars, rods, profiles and wire, nationwide. The potential location is the one that can offer the best level of transit times for Alpha main customers in the northeast US while also can provide greater access to its major terminals in Camp Hill, Lancaster, Harrisburg and York in Pennsylvania. So the alternatives were decided to be Maryland (A1), Pennsylvania (A2), and West Virginia (A3). A decision making team extracted a list of 26 criteria that were urged to affect the relocation decision which were categorized into six main categories: C1: site characteristic, C2: Cost, C3: Traffic access, C4: Marketing opportunity, C5: Quality of living, and C6: Local incentives (see the detailed criteria and subcriteria definition in Ref [10]). This grouping is reasonable, as it was proven that an ordinary decision maker could not handle more than seven or nine decision elements simultaneously without being confused [9].

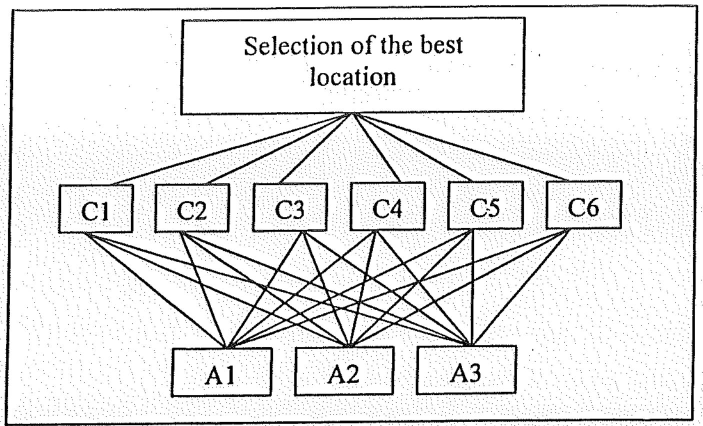


Figure 1: The decision hierarchy for site selection

The complete hierarchy of goal, criteria, and alternatives is constructed (Figure 1), followed by creating the pairwise comparisons of decision criteria and alternatives. The goal of these criteria and the relative weights derived from these pairwise comparisons are shown in Figure 2. These relative weights represent the judgement of decision making team of Alpha on the relative importance of the location criteria with respect to the location decision. Finally in the last step of the AHP process, the relative weights are aggregated to produce a set of synthesized priorities for the three final site alternatives using Expert Choice [6]. Table 1 shows the priority values of these considered alternatives.

Table 1: Location priority alternatives using AHP

Location	Priority	Rank
A1	0.374	1
A2	0.346	2
A3	0.281	3

It is clear that the rank of the alternatives is A1 > A2 > A3. But the differences between priority values are not significant especially between the best alternative and the other less preferred alternatives. Also, the number of pairwise comparisons to be

Step 4: Determine the positive-ideal and negative-ideal solutions.

$$A^* = \{v_1^*, v_2^*, \ldots, v_n^*\}$$
$$= \left\{ \left(\max_i v_{ij} \middle| i \in I'\right), \left(\min_i v_{ij} \middle| i \in I''\right) \right\},$$

$$A^- = \{v_1^-, v_2^-, \ldots, v_n^-\}$$
$$= \left\{ \left(\min_i v_{ij} \middle| i \in I'\right), \left(\max_i v_{ij} \middle| i \in I''\right) \right\},$$

where I' is associated with the benefit criteria, and I" is associated with the cost criteria.

Step 5: Calculate the separation measures, using the n-dimensional Euclidean distance. The separation of each alternative from the positive-ideal solution (Di*) is given as:

$$D_i^* = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - A_j^*\right)^2} \quad i = 1,2,\ldots,m$$

Similarly, the separation of each alternative from the negative-ideal solution (D_i^-) is as follows:

$$D_i^- = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - A_j^-\right)^2} \quad i = 1,2,\ldots,m$$

Step 6: Calculate the relative closeness to the ideal solution and rank the performance order. The relative closeness of the alternative Aj can be expressed as:

$$CC_i^* = \frac{D_i^-}{D_i^* + D_i^-} \quad i = 1,2,\ldots,m$$

where the CC_i^* index value lies between 0 and 1. The larger the index value means the better the performance of the alternatives.

3. Problem Definition

The location selection problem considered in this paper is adopted from the paper of Min [10] where AHP was applied to a case study of a firm (Alpha). This firm plans to move its current manufacturing and distribution facility to a new location in Northeast US. Alpha manufactures and sells chain link fences and their related hardware items

2. Technique for Order Performance by Similarity to Ideal Solution TOPSIS

According to this technique, the best alternative would be the one nearest to the positive ideal solution and farthest from the negative ideal solution [5]. The positive ideal solution is a solution that maximizes the benefit criteria and minimizes the cost criteria, whereas the negative ideal solution is a solution that maximizes the cost criteria and minimizes the benefit criteria. In brief, the positive ideal solution is composed of all best values attainable from the criteria, while the negative ideal solution consists of all worst values attainable from the criteria [13]. There have been lots of studies in the literature using TOPSIS for the solution of MCDM problems [2-4, 7, 12].

The TOPSIS method consists of the following steps:

Step 1: Establish a decision matrix for the ranking. The structure of the matrix can be expressed as follows:

$$\begin{array}{cc} & \begin{array}{ccccccc} F_1 & F_2 & \cdots & \cdots & \cdots & F_n \end{array} \\ A = \begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_i \\ \vdots \\ A_m \end{array} & \begin{array}{cccccc} f_{11} & f_{12} & \cdots & f_{1j} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2j} & \cdots & f_{2n} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ f_{i1} & f_{i2} & \cdots & f_{ij} & \cdots & f_{in} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ f_{m1} & f_{m2} & \cdots & f_{mj} & \cdots & f_{mn} \end{array} \end{array}$$

where A_i denotes the alternatives i, i = 1,2, . . ., m; Fj represents j^{th} criterion, j = 1, 2,. . . , n; and f_{ij} is a crisp value indicating the performance rating of each alternative A_i with respect to each criterion F_j.

Step 2: Calculate the normalized decision matrix R(=[rij]). The normalized value rij is calculated as:

$$r_{ij} = \frac{f_{ij}}{\sqrt{\sum_{j=1}^{n} f_{ij}^2}} \quad i = 1,2,\dots,m;\ j = 1,2,\dots,n.$$

Step 3: Calculate the weighted normalized decision matrix by multiplying the normalized decision matrix by its associated weights. The weighted normalized value vij is calculated as:

$$V_{ij} = w_j \times r_{ij}, \quad i = 1,2,\dots,m;\ j = 1,2,\dots,n.$$

where w_j represents the weight of the j^{th} criterion.

that dominate the current situation scenario, and the preferences of the criteria to be judged.

The main goal of this paper is to demonstrate the usefulness of composing a hybrid location selection model out of a powerful MCDM approaches like AHP, FAHP, and TOPSIS. This hybridization makes use of the advantages of each approach trying to overcome the limitations. Most of the hierarchical decision making approaches such as AHP are only practically usable if the number of criteria and alternatives is sufficiently low because of the large number of performed pairwise comparisons (*m.n.(m-1)/2* pairwise comparison for *m* alternatives and *n* criteria [11]). AHP also lacks the ability to deal with uncertainty associated with vague linguistic criteria comparisons as it uses exact numeric values for judgements. This paper proposes two new hybrid location selection evaluation models. The first proposed hybrid model is the AHP-TOPSIS model. This model is formed of combining the two powerful MCDM models of AHP and TOPSIS. Location alternatives are ranked in AHP-TOPSIS depending on the weights of the AHP in building the TOPSIS weighted evaluation matrix. The concept of creating criteria weights and the evaluation matrix simultaneously saves a lot of time that was consumed in the large AHP pairwise comparisons computations. This model inherits the robustness and flexibility of the AHP. It also makes use of the undeniable advantages of the rational and understood logic of TOPSIS where the computation processes are straightforward and simple. Also, the search of best alternatives for each criterion is presented in a simple and clear mathematical form.

The other developed hybrid model is FAHP-TOPSIS. The weighting of the triangular fuzzy criteria and the evaluations of these criteria are performed simultaneously which in a less computational time than the full FAHP model. FAHP-TOPSIS model is robust, flexible, has simple and straightforward computations, and is understood. Having these advantages of AHP-TOPSIS model, the FAHP-TOPSIS model overcomes the uncertainty insufficiency limitation of the AHP model. It has the ability of handling the vagueness of the data involved in the location selection decision, using triangular fuzzy numbers instead of exact numbers to decide the criteria weights. This paper also demonstrates the effect of weighting the criteria on the final alternatives priority. When the alternatives priorities of equal criteria weights, AHP weights, and FAHP weights are compared, it was apparent that weighting the considered criteria affect the alternatives priority contrast positively. The priority of the best alternative is more contrast to the less preferred alternatives in the FAHP weighted than in the AHP weighted criteria, than in the equal weighted criteria.

The remainder of this paper is structured as follows. Section 2 briefly describes the used TOPSIS method. Section 3, defines the adopted problem of location selection. Section 4 presents the application of the AHP-TOPSIS, while Section 5 demonstrates the FAHP-TOPSIS proposed model in the case of uncertainty. The importance of the criteria weighting stage is demonstrated in section 6. Sections 7 and 8 present conclusions and future work respectively.

MULTI CRITERIA DECISION ASSESSMENT FOR SITE SELECTION USING TWO HOLISTIC HYBRID MODELS

Mohamed A. Shouman[1], Mahmoud Abd El-Moaty[2], Eman S. Hasan[3]

Abstract: *Selecting a particular location is a multi criteria decision making (MCDM) problem which is subjected to multiple conflicting tangible and intangible criteria. Some tangible criteria are considered such as land cost, building construction cost, traffic access, site specific features, market opportunity. Other intangible criteria as climate, pollution, living expense, and traffic congestion are also considered. Two newly hybrid evaluation models for site selection are proposed in this paper. The first proposed AHP-TOPSIS model combines the analytical hierarchy process (AHP) with the technique for order performance by similarity to ideal solution (TOPSIS) having the advantage of eliminating a large number of computations that is required if the AHP evaluation model is used. However when it is difficult for the decision maker to express preferences using exact numerical values, it is more desirable to use fuzzy evaluations to handle the vagueness of the data involved in the location selection decision. That was developed in the FAHP-TOPSIS proposed model where fuzzy analytical hierarchy process (FAHP) is combined with TOPSIS to handle data uncertainty. A case study adopted from previous literature is discussed to show the appropriate application and usefulness of the proposed models for potential use in site selection decision over the other existing evaluation models.*

Keywords: Site selection, Multi Criteria Decision Making (MCDM), AHP, FAHP, TOPSIS.

1. Introduction

Selecting a particular location is a multi criteria decision making (MCDM) problem which is subjected to multi conflicting tangible and intangible criteria. Some of the locations under consideration might be good with respect to some criteria, but poor under other criteria. Although several methods exist for MCDM, there are no better or worse techniques. But some techniques better suit to particular decision problems than others do [8]. The MCDM method should not be chosen until the analyst and the decision makers: a) understand the problem, b) declare the feasible alternatives, c) realize the different outcomes, d) understand the conflicts between the criteria, and e) understand the level of data uncertainty. Certain locations should be evaluated on the basis of a certain set of criteria, and the outstanding location should be selected. Some of the questions are generally asked in location selection problem to know the criteria

[1] Professor of Operations Research and Decision Support, Decision Support Department, Faculty of Computers and Informatics, Zagazig University.

[2] Lecturer in Decision Support Department, Faculty of Computers and Informatics, Zagazig University.

[3] Demonstrator in Decision Support Department, Faculty of Computers and Informatics, Zagazig University.

programming. European Journal of Operational Research, 171, 811-829.

13. Gwee, B. H. & Lim, M. H. (1996). Polyominoes tiling by a genetic algorithm. Computational Optimization and Applications, 6, 273-291.

14. Jain, S. & Gea, H. C. (1998). Two-Dimensional packing problems using genetic algorithms. Engineering with Computers, 14, 206-213.

15. Li, Z. (1994). Compaction algorithms for non-convex polygons and their applications. Ph.D. thesis, Harvard University.

16. Manabe, S. & Asai, H. (2001). A neuro-based optimization algorithm for tiling problems with rotation. Neural Processing Letters, 13, 267-275.

17. Segenreich, S. A. & Braga, L. M. (1986). Optimal nesting of general plane figures: A Monte Carlo Heuristical Approach. Computer & Graphics, 10(3), 229-237.

18. Takefuji, Y. & Lee K.C. (1990). A parallel algorithm for tiling problems. IEEE Transactions on Neural Networks, 1 (1), 143-145.

2. Amintoosi, M., Yazdi, H. S. b, Fathy, M. & Monsefi, R. (2007). Using pattern matching for tiling and packing problems. European Journal of Operational Research, 183, 950-960.

3. Babu, A. R. & Babu, N .R. (2001). A generic approach for nesting of 2-D parts in 2-D sheets using genetic and heuristic algorithms. Computer-Aided Design, 33, 879-891.

4. Bennell, J. A. & Dowsland, K. A. (1999). A tabu thresholding implementation for the irregular stock cutting problem. Int. J. Prod. Res., 37(18), 4259- 4275.

5. Burke, E., Hellier, R., Kendall, G. & Whitwell, G. (2006). A new bottom-left-fill heuristic algorithm for the two dimensional irregular packing problem. Operations Research, 54(3), 587-601.

6. Daniels, K. K., Milenkovic, V. J. & Li, Z. (1994). Multiple containment methods. Technical Report 12-94, Center for Research in Computing Technology, Harvard University, Cambridge.

7. Egeblad, J., Nielsen, B. K. & Odgaard, A. (2007). Fast neighborhood search for two and three-dimensional nesting problems. European Journal of Operational Research, 183, 1249-1266.

8. Fischer, A. D. & Dagli, C. H. (2004). Employing subgroup evolution for irregular-shape nesting. Journal of Intelligent Manufacturing, 15, 187-199.

9. Fischetti, M. & Luzzi, I. (2008). Mixed-Integer programming models for the nesting problem. J Heuristic, DOI 10.1007/s10732-008-9088-9.

10. Fowler, R. J., Paterson, M. S. & Tanimoto, S. L. (1981). Optimal packing and covering in the plane are NP-complete. Information Processing Letters, 12 (3), 133-137.

11. Golomb, S.W. (1954). Checkerboards and Polyominoes. The American Mathematical Monthly, 61 (10).

12. Gomes, A. M. & Oliveira. J. F. (2006). Solving irregular strip packing problems by hybridising simulated annealing and linear

Solution times are not reported by the authors, but another vital performance measure is reported. It is the optimum achievement-rate, which is stated among 10 solution runs for each dataset. Accordingly, optimum achievement-rates that range between 40% and 100% with an average of 76% are reported.

5. Conclusion

A simple MIP model is formulated for solving both the strip packing and the tiling problems that involve special irregular-shaped items known as polyominoes. Through the preliminary tests carried out on an average PC, the proposed model proved to handle problem sizes of about 3 million total variables e.g. 45 polyominoes inside a 10X25 sheet, or 8 polyominoes inside a 20X30 sheet), while at higher problem sizes, a physical memory shortage is expected. Moreover, almost the solution time is reasonable, where it mainly depends on the size of the associated number of integer variables. Approximate simple formulae are derived for rough estimation of the size of a given problem in terms of the practical parameters. The proposed model is verified on the benchmark datasets collected from the literature for each of the addressed problems. Despite tighter rotation constraints are held by the model it always, except for one case, reaches the same optimal reported results. Finally, the proposed model is regarded as a practical tool for tackling the mentioned problems with the least programming effort, while it has a high potential for being modified so as to handle almost all polyominoes-related cutting and packing problems with different objectives, assumptions and considerations.

Future work would modify the proposed model so as to allow for additional rotation angles as well as reflection, in addition to adopting different packing objectives such as the one used in Jain and Jea (1998). Other problem considerations such as using irregular sheets, sheets with defects, and multiple sheets could be included. Finally, the proposed model could be employed in building approximate patterns for the general irregular packing problems.

List of References

1. Albano, A. & Sapuppo, G. (1980). Optimal allocation of two-dimensional irregular shapes using heuristic search methods. IEEE Transactions on Systems, Man, and Cybernetics, 10(5), 242-248.

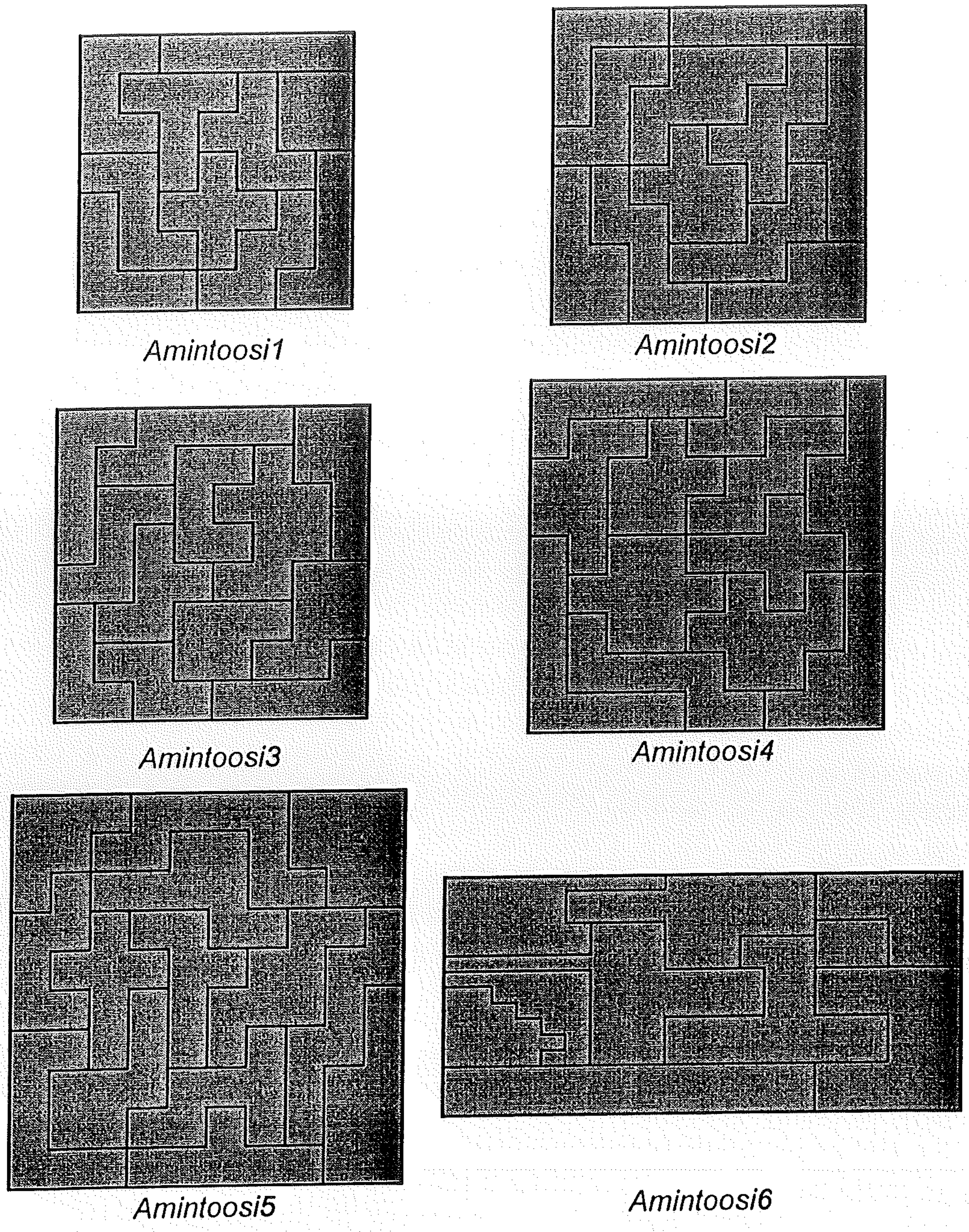


Figure 4. Obtained patterns for the tiling benchmark datasets

The obtained patterns for the entire problems are exactly the same for those reached by Amintoosi et al. (2007), except for Amintoosi4, and Amintoosi6. As for Amintoosi4, the pattern obtained by the model is considered to be the one obtained by Amintoosi et al. (2007) but entirely rotated by 180°, the situation is the same for Amintoosi6 in addition to minor positions-exchanges between polyominoes.

Babu and Babu (2001) reach the same minimum length with a different solution layout. The proposed model reached the same minimum lengths obtained by Jain and Gea (1998) in Jain1 and Jain3. For Jain2, the model obtained an optimal solution that is one unit length larger since Jain and Gea (1998) assumed that the 90° and 270° rotation of polyominoes are allowed. It is noticed in the last three patterns (Jain1, Jain2, and Jain3) that the model always tends to shift the gaps towards the open end of the sheet as much as it can, forced by the penalty-based objective function. Regarding the computation time, Jain and Gea state that all the reported solutions are reached in less than 30 minutes on a Sparc 5 Sun Station.

4.3 Polyominoes Tiling

Much interest in the literature is noticed for the polyominoes tiling problem, and more problems are found for comparison purposes. However, the six benchmark datasets addressed in Amintoosi et al. (2007) would be adequate, since they are the most recent known to us. Additional datasets that are very similar to the mentioned ones could also be found in Gwee and Lim (1996). The general characteristics of these datasets are shown below in Table 3.

Table 3
Main characteristics of the strip packing benchmark datasets

Dataset	N	W	L	S	$\bar{s}$	$\bar{p}$	Total no. of variables	No. of integer variables	No. of constraints
Amintoosi1	10	7	7	49	20	4.9	24,990	522	2031
Amintoosi2	11	8	8	64	25	5.82	46,464	716	3,488
Amintoosi3	12	8	8	64	20	5.33	50,688	878	3,537
Amintoosi4	12	9	9	81	25	6.75	80,676	988	5,868
Amintoosi5	14	10	10	100	36	7.14	142,800	1,512	9,155
Amintoosi6	10	15	21	315	135	31.5	998,550	2,774	57,381

The default zero-orientation of each polyomino throughout the mentioned datasets is assumed to be the one by which each polyomino is found in the solution patterns presented by Amintoosi et al. (2007), as they are not stated. The patterns obtained through the proposed model are presented in Figure 4, where each of them is obtained in less than 3 minutes.

for the sheet length. Table 2 presents the main characteristics of the four benchmark problems.

Table 2
Main characteristics of the strip packing benchmark datasets

Dataset	N	W	L	S	$\bar{s}$	$\bar{p}$	Total no. of variables	No. of integer variables	No. of constraints
Babu	14	8	12	96	16	2.29	131,712	2,404	3,119
Jain1	26	10	8	80	4	3	170,560	3,276	6,659
Jain2	19	10	8	80	6	4	157,320	2,394	7,292
Jain3	42	12	10	120	6	2.81	742,896	9,380	16,595

The obtained patterns through the proposed model are presented in Figure 3, where each of them is obtained in less than 2 minutes.

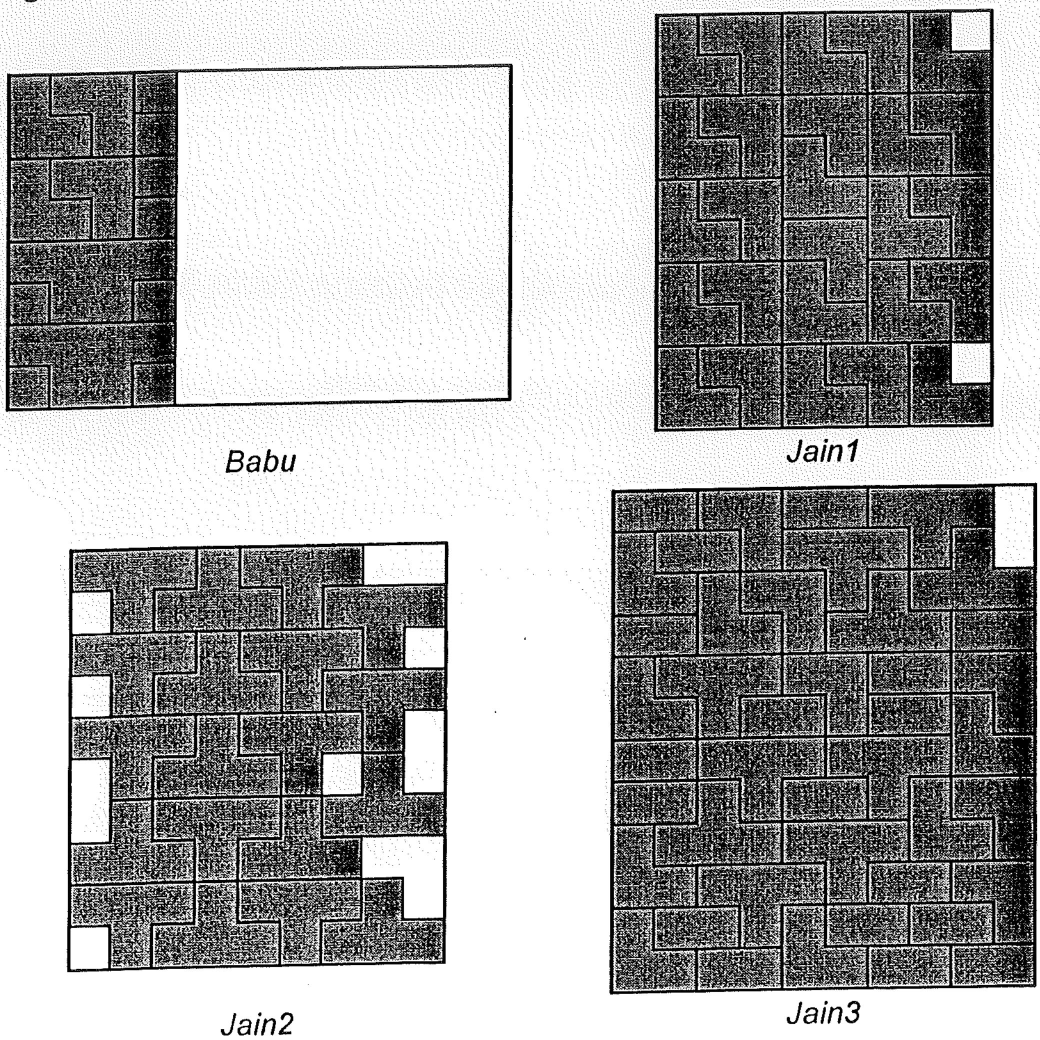


Figure 3. Obtained patterns for the strip packing benchmark datasets

Table 1
Exact sizes for a set of selected problem instances

Average polyo. size $\bar{s}$	Average no. of cells $\bar{p}$	Sheet size S	No. of polyo. N	Total no. of variables	No. of integer variables	No. of constraints	Time (s)
6	*4*	*260*	*42*	2,861,040	18,144	54,735	223
8	*6*	*280*	*35*	2,763,600	15,750	63,316	201
10	*8*	*300*	*30*	2,718,000	14,040	70,531	241
12	*10*	*330*	*27*	2,958,120	13,608	82,006	313
14	*12*	*340*	*24*	2,790,720	12,096	85,037	76
16	*14*	*360*	*22*	2,867,040	11,484	92,255	167
18	*16*	*360*	*20*	2,606,400	10,080	91,101	55
20	*18*	*380*	*19*	2,758,040	9,918	99,580	66
.							
54	*52*	*550*	*10*	2,926,800	5,040	136,631	169

As long as the total number of variables is almost below 3 million, the maximum memory limits are not reached (no physical memory shortage). Thus, it could be concluded that for the proposed formulation, the memory limits depends only on the total number of variables regardless the number of integer variables or constraints. Moreover, the number of integer variables has the greatest impact, among the three problem-size elements, on the solution times. It is also clear that up to the previously mentioned memory limit (about 3 million variables); there are no troubles with the solution time, which is expected to arise if the available physical memory is augmented. Further tests that are not reported here revealed that after about 20,000 integer variables, the solution times spent by the model would be unreasonable.

4.2 Polyominoes Strip Packing

Four published benchmark problems are used to verify the proposed model. A single polyominoes strip packing problem is found in Babu and Babu (2001), while three other problems that can be treated as a polyominoes strip packing problems are in Jain and Gea (1998). The problem addressed is to arrange a given set of polyominoes into the form that minimizes their entire rectangular enclosure. Hence, we will consider the vertical dimensions of their presented patterns as the sheet width for a supposed strip packing problem, and the corresponding vertical dimensions as an estimate problem is related to the total number of variables, it is concluded that the initial sheet length, during handling a strip packing problem, should be estimated carefully. Additionally, the model could manipulate a high number of polyominoes if only they have a small average size, or correspondingly a small sheet size is needed. On the other hand, the model is expected to manipulate a problem with large sheet size or correspondingly high average polyominoes size when only a small number of polyominoes is incorporated.

4. Experimental Results

Through this section, three sets of experiments are performed; the first set involves preliminary tests carried out in order to roughly quantify the relation between a given problem size and the memory limit, in addition to the solution time. In the second set of experiments, the proposed model is tested on the polyominoes strip packing benchmark datasets, while in the third set of experiments the model is tested on the polyominoes tiling benchmark datasets. The results obtained through the last two sets of experiments are compared with those found in the literature. LINGO 8.0 was utilized throughout the conducted experiments, using a laptop of 1.5 GB RAM supported by an Intel® Core Duo processor of 1.66GHz CPU speed and 2MB Cache Size, operated with a Microsoft Windows XP Professional O/S.

4.1 Computational Time and Memory

Several problems with different configurations were generated, each contains a set of identical polyominoes. The basic difference between those problems is the average polyominoes size $\bar{s}$. In each case, the number of polyominoes is determined so that the maximum memory limits are about to be reached. That is to say that all the considered problem sizes are just about to hit the memory borders (physical memory shortage). Therefore, both the memory limit along with the corresponding computational time could be correlated with the problem size. A common sheet width of ten cells is utilized. For estimating an upper limit on the employed sheet lengths, It is assumed that the expected utilization (u) is given by $\bar{p}/\bar{s}$, where $\bar{p}$ is the average number of cells per polyomino. Subsequently, the estimated length for the associated patterns should be $N\bar{p}/Wu$ which is equivalent to $N\bar{s}/W$. A summary of the results is presented in Table 1.

3.2. Problem Size Analysis

An important issue concerning the formulation presented in the previous subsection is the large number of associated decision variables, which consequently corresponds to high demand of physical memory and computational time needed to handle a given problem. The relation between the different model parameters and the problem size as represented by the number of total variables, integer variables, and constraints are given by:

Number of constraints $:\sum_{k=1}^{n} U_k V_k w_k l_k + WL + N$ (7)

Total number of variables $: NW^2L^2 + 2NWL$ (8)

Number of integer Variables $: 2\sum_{k=1}^{n} U_k V_k$ (9)

The following approximations may be useful to characterize the problem size in terms of the practical parameters. A polyomino width w_k and a polyomino length l_k are represented by the mean width $\bar{w}$ and the mean length $\bar{l}$. Moreover the farthest possible position for a given polyomino along the width direction of the sheet U_k is always approximated by the sheet width W, and similarly V_k is approximated by the sheet length L. Obviously, the last approximation is more realistic at small values for $\bar{w}/W$ and $\bar{l}/L$. Upon applying those approximations in addition to neglecting the relatively small terms, the following simpler formulae are obtained, where the term WL is given the symbol S, denoting the sheet size, while $\overline{wl}$ is given the symbol $\bar{s}$, denoting the average polyomino size. The approximate formulae could be written as follows:

Number of constraints $: N\,S\bar{s}$ (10)

Total number of variables $: N\,S^2$ (11)

Number of integer Variables $: 2\,N\,S$ (12)

Thus, it may be deduced that each of the number of polyominoes N, the sheet size S, and the average polyomino size $\bar{s}$, all have the same impact on the number of constraints. Also, both N and S have an equal effect on the number of integer variables. The total number of variables is highly sensitive to the sheet size S as expected. As the required physical memory to handle a given

The complete form of the proposed mathematical formulation is given as:

$$Minimize \sum_{k=1}^{N} \sum_{i=1}^{U_k} \sum_{j=1}^{V_k} \sum_{m=1}^{W} \sum_{n=1}^{L} S_{k,i,j,m,n}\, n \qquad (1)$$

Subject to

$$S_{k,i,j,m+i-1,n+j-1} - X_{k,i,j}\, P_{k,m,n} - Y_{k,i,j}\, P_{k,w_k-m+1,l_k-n+1} = 0 \qquad \forall k = 1, \dots, N,\ \forall i = 1, \dots, U_k,\ \forall j = 1, \dots, V_k,\ \forall m = 1, \dots, w_k,\ \forall n = 1, \dots, l_k \qquad (2)$$

$$\sum_{i=1}^{U_k} \sum_{j=1}^{V_k} X_{k,i,j} + \sum_{i=1}^{U_k} \sum_{j=1}^{V_k} Y_{k,i,j} = 1 \qquad \forall k = 1, \dots, N \qquad (3)$$

$$\sum_{k=1}^{N} \sum_{i=1}^{U_k} \sum_{j=1}^{V_k} S_{k,i,j,m,n} \le 1 \qquad \forall m = 1, \dots, W,\ \forall n = 1, \dots, L \qquad (4)$$

$$X_{k,i,j} \in \{0,1\} \qquad \forall k = 1, \dots, N,\ \forall i = 1, \dots, U_k,\ \forall j = 1, \dots, V_k, \qquad (5)$$

$$Y_{k,i,j} \in \{0,1\} \qquad \forall k = 1, \dots, N,\ \forall i = 1, \dots, U_k,\ \forall j = 1, \dots, V_k \qquad (6)$$

Another objective function that stands for the produced waste was investigated; however the one introduced above has proven to be more efficient in terms of solution time. For estimating the sheet length for a given strip packing dataset, a quick approach is to roughly expect the utilization percent according to the irregularity extent or any other criterion, and correspondingly the sheet length is calculated. A more effective approach is to use a simple ordering rule (e.g. decreasing area) along with a traditional placement procedure such as the scanning procedure discussed in Segenreich and Braga (1986) for generating an initial feasible solution, in which the obtained length of such a solution would be a realistic estimate for the sheet length.

This set is responsible for generating the set of all possible placement arrays for each polyomino. The second main set of constraints is given by:

$$\sum_{i=1}^{U_k}\sum_{j=1}^{V_k} X_{k,i,j} + \sum_{i=1}^{U_k}\sum_{j=1}^{V_k} Y_{k,i,j} = 1 \qquad \forall\, k = 1, \dots, N$$

This set guarantees that exactly a single placement array is only considered for each polyomino. The final main set of constrains is overlap prevention constraints and is stated as follow:

$$\sum_{k=1}^{N}\sum_{i=1}^{U_k}\sum_{j=1}^{V_k} S_{k,i,j,m,n} \le 1 \qquad \forall\, m = 1, \dots, W, \forall\, n = 1, \dots, L$$

This set of constraints sums the entire placement arrays and ensures that neither of the obtained cells is occupied by more than 1, given that the unselected placement arrays are always null arrays. Another set of constraints are applied to force the variables $X_{k,i,j}$ and $Y_{k,i,j}$ to assume binary values. A penalty based objective function is developed and is given by:

$$Minimize \sum_{k=1}^{N}\sum_{i=1}^{U_k}\sum_{j=1}^{V_k}\sum_{m=1}^{W}\sum_{n=1}^{L} S_{k,i,j,m,n}\, n$$

The proposed function penalizes each cell within the array resulting from summing the entire placement arrays. The penalty value for a given cell is set according to the sheet length where such a cell is located. Hence, the objective value for a given solution is given by the total sum of penalties. Therefore, the utilized sheet length will always be forced to be short as possible, so as to have the minimum value for the employed objective function. Figure 2 graphically demonstrates the idea.

1	1	0	0	1	0	0	1
0	1	1	1	1	1	1	1
0	1	1	0	1	0	1	0
1	1	1		0	0	0	0

Array resulted from summing the entire placement arrays

1	2	0	0	5	0	0	8
0	2	3	4	5	6	7	8
0	2	3	0	5	0	7	0
1	2	3		0	0	0	0

Corresponding penalty values

Figure 2. Graphical demonstration for the proposed objective function

in Figure 1 could be considered as a placement array if its corresponding shaded cells are filled by 1 and the other cells are filled by 0. Hence the model has to generate all the possible placement arrays and select only a single placement array for each polyomino. The model input parameters are stated as follow:

N	No. of involved polyominoes
W	Sheet width
L	Sheet length (an upper limit that should contain all the items)
w_k	Width of the polyomino k
l_k	Length of the polyomino k
$P_{k,m,n}$	A parameter that represents the cell m,n within the binary array encoding the polyomino k at its default orientation, in which it is filled by 1 if occupied by any of the polyomino cells, and 0 otherwise
U_k	Farthest possible position for the polyomino k along the width direction of the sheet $= W - w_k + 1$
V_k	Farthest possible position for the polyomino k along the length direction of the sheet $= L - l_k + 1$

The following decision variables are also defined:

$X_{k,i,j}$	A binary variable that takes the value of 1 if the cell i, j is selected to be the placement position of the polyomino k, oriented at 0° orientation
$Y_{k,i,j}$	A binary variable that takes the value of 1 if the cell i, j is selected to be the placement position of the polyomino k, oriented at 180° orientation
$S_{k,i,j,m,n}$	A binary variable that stands for the cell m, n within a placement array that corresponds to the assignment of polyomino k to the position i, j

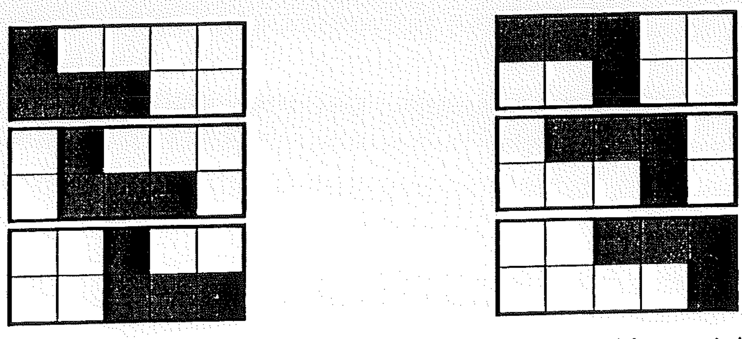
Three main sets of constraints are developed; the first one is stated as:

$$S_{k,i,j,m+i-1,n+j-1} - X_{k,i,j}\, P_{k,m,n} - Y_{k,i,j}\, P_{k,w_k-m+1,l_k-n+1} = 0 \qquad \begin{array}{l} \forall\, k = 1, \dots, N, \\ \forall\, i = 1, \dots, U_k, \forall\, j = 1, \dots, V_k, \\ \forall\, m = 1, \dots, w_k, \forall\, n = 1, \dots, l_k \end{array}$$

integer multiples of the same basic unit length, that defines the polyominoes' dimensions. Henceforth, the sheet length is assumed to be along the horizontal direction, and its origin is the bottom left corner.

3.1 Mathematical Formulation

The proposed model is based on the representation of a given sheet in the form of a grid of square cells, where a polyomino is also a cluster of square cells. Thus, the sheet area is reduced into a finite set of placement points, where each point stands for a square cell. Accordingly, the model task is then to select out of this set of points the placement position of each polyomino, as well as to select its corresponding orientation (0° or 180°). The main objective is to minimize the used sheet length, while neither of the polyominoes cells is overlapped. A polyomino position is defined by the position of its reference cell which is the upper left one. Only positions that allow for the whole polyomino to be contained within the sheet borders are considered by the model. Figure 1 exhibits a graphical representation for all the possible positions for a 4-cells polyomino, at 0° and 180° rotation angles, inside a sheet of 2 cells width and 5 cells length.



Possible positions at 0° Possible positions at 180°

Figure 1. Graphical representation for possible placements for a given polyomino

In order to mathematically formulate the above approach, the geometry of each polyomino is encoded by a two-dimensional binary array, which is one of the model input parameters. A separate two-dimensional array, *placement array*, with the same size of the sheet is used to signify a polyomino at a given position and orientation. In which each placement array is occupied only by the elements of the associated polyomino's binary array. Each of the six arrays presented single dataset reveal the effectiveness of the proposed algorithm over that of Jain and Gea (1998).

The published approaches for polyominoes tiling problems are mainly either a neuro-based or evolutionary-based algorithms. In Takefuji and Lee (1990), a tiling algorithm that do not allow for polyominoes rotation or reflection is introduced. The authors employ a Hopfield neural network with optimization approach. The algorithm is tested on a 7X7 sheet involving 10 polyominoes and upon adjusting the algorithm parameters; it was capable of reaching an optimum solution in 100% of its runs. Such an algorithm is extended by Manabe and Asai (2001), so that to additionally handle rotatable polyominoes. The modified algorithm is tested on five datasets; three of them involve a 5X5 sheet with 5 polyominoes and the other two involve 6X6 sheet with 6 polyominoes. An average optimum achievement-rate of 57% is reported. An order-based genetic algorithm is introduced by Gwee and Lim (1996), along with a proposed circular placement strategy that deals with both, rotation and reflection. Four datasets ranging from 8X8 sheet with 8 polyominoes to 32X32 sheet with 128 are tested. An optimal configuration is obtained, while the corresponding achievement-rates are not reported. In Amintoosi et al. (2007), an evolutionary algorithm is implemented to search for the polyominoes placement sequences as well as their associated orientations, where a string matching based tiling procedure generates patterns correspondingly. Six datasets were tested, with the larger one consists of a 15X21 sheet and 10 polyominoes, in which an average optimum-achievement rate of 76% is obtained.

Therefore all the existing solution approaches are either an impractical exact approaches, or a heuristic approaches that do not guarantee an optimum solution.

3. Proposed Mixed-Integer Model

A new formulation is introduced for the polyominoes strip packing problem. The same formulation can readily handle the polyominoes tiling problem as well, given that at least one perfect pattern exists. The proposed model is a mixed-integer model that allows polyominoes to contain internal holes, and permits a polyomino to rotate by 180°, but does not permit reflecting a polyomino along an axis. Moreover, it is assumed that the sheet dimensions are also

In this paper a new mathematical formulation that is capable of handling polyominoes strip packing and tiling problems is introduced. The problem sizes (no. of polyominoes and sheet dimension) by which the model could manipulate is analyzed. Furthermore, practical implementation of the proposed model is carried out on a set of problems, including benchmark datasets collected from the literature. In the next sections, the previous work is reviewed and the proposed mathematical model is constructed and tested. The model proved to handle reasonable problem sizes (e.g. 45 polyominoes inside a 10X25 sheet, or 8 polyominoes inside a 20X30 sheet) and reached the optimal solutions in each of the considered benchmark datasets.

2. Literature Review

Due to the strong NP-hardness nature of the irregular packing problems (Fowler et al. (1981)), they are used to be tackled practically through two main heuristic-based approaches; the first one, direct approach, builds an initial pattern that shouldn't be a feasible one, in which a search technique is directly applied to the items positions to reach a good feasible pattern (Bennell and Dowsland (1999), Gomes and Oliveira (2006), and Egeblad et al. (2007)). While the second approach, sequence-based approach, searches for a good packing sequence to be employed under certain placement procedure (Albano and Sapuppo (1980), Fischer and Dagli (2004), and Burke et al. (2006)). Regarding the mathematical approaches, based on previous proposals of Daniels et al. (1994), and Li (1994), Fischetti and Luzzi (2008) describe a mixed-integer programming model that makes use of the no-fit polygon overlap detection tool. Such a model is found to be the only one to entirely handle an irregular packing problem. However it is not capable of handling more than nine polygons, in a reasonable solution time.

Concerning polyominoes packing problems specifically, Jain & Gea (1998) deal with the problem of packing a set of polyominoes, so that the rectangular sheet area required to enclose all the considered polyominoes is minimized. A genetic algorithm is employed to search for polyominoes positions and orientations. The tests conducted via three problems provide final patterns with only few gaps. In Babu and Babu (2001), the authors additionally allow for polyominoes reflection and a genetic algorithm is also utilized along with a proposed scanning-based placement procedure. Comparisons performed on a

A Mixed-Integer Model for Two-Dimensional Polyominoes Strip Packing and Tiling Problems

Mohamed N. Kashkoush, Mohamed A. Shalaby and Ehab A. Abdelhafiez

Mechanical Design and Production Dept., Faculty of Engineering, Cairo University, Giza, Egypt 12613

Abstract

In this paper, a new Mixed-Integer Programming model is introduced for handling a special case of the 2-D irregular strip packing problem, in which each of the involved items is a collection of identical square cells connected along their edges, named polyomino. Besides, the model could handle a rectangle tiling problem with polyominoes. The problem sizes by which the model could manipulate are investigated, and finally the proposed model is verified on the benchmark datasets collected from the literature for each of the mentioned problems.

Keywords: Mixed-Integer Programming; Polyominoes; Tiling; Irregular Strip Packing

1. Introduction

The classical two-dimensional irregular strip packing problem is such that for a given set of items of irregular shapes, and a rectangular sheet of fixed width and an open length; it is required to assign each item a distinct position, in a non overlapping manner, so that the utilized length of the sheet (generated pattern length) is minimized. A variant of this problem is the one where the geometry of the involved items is restricted to rectilinear polygons, known as, polyominoes. A *polyomino,* first introduced by Golomb (1954), is special polygon with all of its edges are orthogonal lines, and each line length is integer multiples of a basic unit length. The polyominoes strip packing problem is classified as a polyominoes tiling problem if there exists a given sheet length for which the utilization is 100%. Typical applications for polyominoes packing problems in general could be found in the sheet metal stamping, and in the design of integrated circuits.

6. References:

[1] Bazaraa. M. S., "Nonlinear Programming Theory and Algorithms", John wiley, Sons New York, (1979).

[2] G P Liu, J B Yang, and J F Whidborne, "Multiobjective Optimisation and Control", Research Studies Press Ltd, 2003.

[3] Jin Liangchao, Gu Jifa and Shu Guangfu, "Order Number Methods for MCDM", J. Sys Sci and Math Scis, 4(4), P. 281-294, (1984).

[4] Kaisa Miettinen, "Nonlinear Multiobjective Optimization", Kluwer Academic Publishers, Fourth Printing 2004.

[5] Lasdon. L. S., "Optimization Theory for Large Scale Systems", Macmillan Company, New York, (1970).

[6] Osman. M. S. A., "Qualitative Analysis of Basic Notions in Parametric Convex Programming I, (Prameters In The Constraints)", Appl. Math. Cssr Akad. Ved. Prague, 22, P. 318-332, (1977).

[7] S.V. Utyuzhnikov , P. Fantini , and M.D. Guenov, "A method for generating a well-distributed Pareto set in nonlinear multiobjective optimization", Journal of Computational and Applied Mathematics 223 (2009) 820–841.

4. Solving the following problem to find the best efficient solution:

$$\text{Minimize} \sum_{i=1}^{3} \eta_i$$

Subject to

$$(x, y) \in \bar{M}^{o}$$

The solution of the above problem is the best efficient solution according to η-measurement.

For clarifying the previous situation, let us make a comparison between some of the efficient solutions and other feasible solutions in this example as follows:

	f_1	f_2	$\sum_{i=1}^{3} \eta_i$
Efficient $X_1 = (1, 2)$	1	2	1.5
Efficient $X_2 = (2.5, 0)$	2.5	0	2
Efficient $X_3 = (2, 1)$	2	1	1.5
Efficient $X_4 = (2.1, 0.8)$	2.1	0.8	1.6
Efficient $X_5 = (1.5, 1.5)$	1.5	1.5	1.5
Weakly efficient $X_6 = (0, 2)$	0	2	2.5
Inferior $X_7 = (1, 1)$	1	1	2.5

5. Conclusion

In this paper, a treatment for finding all efficient solutions by using a modified ε-constraint technique which gives the set of all efficient solutions. Also, the proposed method is illustrated to find the best efficient solution among the efficient solutions where there is no priority among the objectives of the multiobjective programming problem. The η-measurement principle for multiobjective programming problem is introduced for ranking the efficient solutions and for finding the best efficient solution.

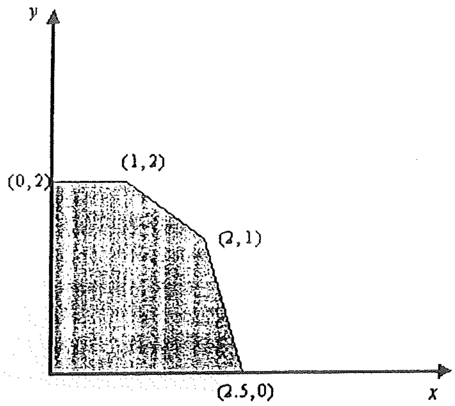


Fig. 1, the set of feasible region

2. Solving the ε-constraint problem as described in the algorithm section in step (2) and step (3). The sets of all weakly efficient solutions according to each ε-constraint problem is illustrated in figure 2 by the bold line:

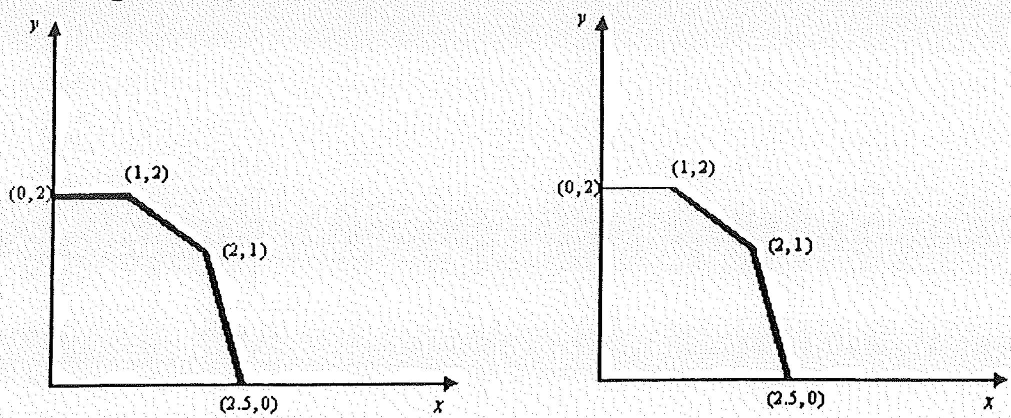


Fig. 2, the set of weakly efficient solutions

3. Finding the set of all efficient solutions which is denoted by $\bar{M}^{o}$. Where $\bar{M}^{o}$ is illustrated in figure 3 by the bold line:

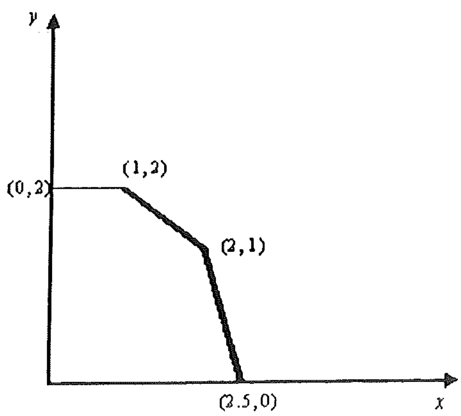


Fig. 3, the set of efficient solutions

According to the previous data, $X_3 = (0.75, 1.25)$ is the best efficient solution.

Remark:

1. The η-measurement principle for multiobjective programming problem can be easily used to arrange the efficient solution
2. The η-measurement principle can be used as a new method for dominance between efficient solutions and between efficient solutions and weakly efficient solution

4.2.Second Example

Solve the following multiobjective programming problem using the proposed algorithm in section (3).

$$\text{Maximize } F(X) = \{f_1, f_2\}$$

$$\text{Subject to}$$

$$\bar{M} = \left\{ X \in R^2 \left| \begin{array}{l} x + y \leq 3 \\ x + 0.5y \leq 2.5 \\ y \leq 2 \\ x \geq 0 \\ y \geq 0 \end{array} \right. \right\}$$

Where

$f_1 = x$

$f_2 = y$

Solution:

1. Finding the ideal point by solving each objective function separately. Then the optimal values are $f_1^o = 2.5$, and $f_2^o = 2$. Figure 1 illustrates the feasible region.

Solution:

1. Finding the ideal point by solving each objective function separately. Then the optimal solutions are $X_1^* = (1,1)$, $X_2^* = (0.5, 1.5)$, and $X_3^* = (0.75, 1.25)$ and the optimal values are $f_1^* = 0$, $f_2^* = 0.5$, and $f_3^* = 0.125$ respectively.
2. Solving the ε-constraint problem as described in the algorithm section in step (2) and step (3). The sets of all weakly efficient solutions according to each ε-constraint problem is given by:
   - $\bar{M}^o_{\eta 1} = \{(x,y) | (x,y) = \lambda(1,1) + (1-\lambda)(0.5,1.5)\}$
   - $\bar{M}^o_{\eta 2} = \{(x,y) | (x,y) = \lambda(1,1) + (1-\lambda)(0.5,1.5)\}$
   - $\bar{M}^o_{\eta 3} = \{(x,y) | (x,y) = \lambda(1,1) + (1-\lambda)(0.5,1.5)\}$

   Remark:

   $0 \le \eta_i$, for each $i = 1, 2, 3$
3. Finding the set of all efficient solutions which is denoted by $\bar{M}^o$. Where:

$$\bar{M}^o = \bigcap_{w=1}^{3} \bar{M}^o_{\eta w} = \{(x,y) | (x,y) = \lambda(1,1) + (1-\lambda)(0.5,1.5)\}$$

4. Solving the following problem (η-measurement problem) to find the best efficient solution:

$$\text{Minimize } \sum_{i=1}^{3} \eta_i$$

Subject to

$$(x,y) \in \bar{M}^o$$

The solution of the above problem is the best efficient solution according to η-measurement.

For clarifying the previous situation, let us make a comparison between some of the efficient point in this example as follows:

	$X_1 = (1,1)$	$X_2 = (0.5, 1.5)$	$X_3 = (0.75, 1.25)$	$X_4 = (0.8, 1.2)$
f_1	0	0.5	0.125	0.08
f_2	1	0.5	0.625	0.68
f_3	0.25	0.25	0.125	0.13
$\sum_{i=1}^{3} \eta_i$	0.625	0.625	0.25	0.265

Step (3):

Repeat step (2) by exchanging the objective function f_w by another one and find the corresponding set of the weakly efficient solutions until all set of the weakly efficient solutions for all the objectives are determined.

Step (4):

Finding the set of all efficient solutions $\bar{M}^o$. Where:

$$\bar{M}^o = \bigcap_{w=1}^{r} \bar{M}^o_{\eta w}$$

Step (5):

Solve the i^{th} sub-problem given by:

$$\text{Minimize } \eta = \sum_{i=1}^{r}\left(f_i(x) - f_i^o\right)$$

Subject to

$x \in \bar{M}^o$

$\eta_i \geq 0$

Remark:

1. The optimal solution of the problem that is described in step (5) represents the best efficient solution among the efficient solutions for the multiobjective programming problem (1).
2. The problem in step (5) is called η-measurement principle.

4. Illustrative Examples

4.1. First Example

Solve the following nonlinear multiobjective programming problem using the described algorithm in section (3).

Minimize $F(X) = \{f_1, f_2, f_3\}$

Subject to

$M = \{X \in R^2 \mid x + y = 2, x \geq 0, y \geq 0\}$

Where

$f_1 = (x-1)^2 + (y-1)^2$

$f_2 = (x-1)^2 + (y-2)^2$

$f_3 = (x-0.5)^2 + (y-1)^2$

Where

$$m_{opt}(\eta') = \left\{ x' \middle| f^{o}_{\eta_i} = \text{minimize} f_i(x), \text{s.t } x \in \overline{M}_{\eta_i}(\eta') \right\}$$

Definition 2.3 " stability set of the first kind with a balanced number η "

Let $\eta' \in SV_{\eta}$ with a corresponding optimal point x^*, the stability set of the first kind with a balanced number η, $S^{(1)}_{\eta}(x^*)$, of problem (2) corresponding to x^* is defined by:

$$S^{(1)}_{\eta}(x^*) = \left\{ \eta' \in R^m \middle| x^* \in \bigcap_{i=1}^{r} \overline{M}_{\eta_i}(\eta') \right\}$$

3. The Algorithm

The algorithm of solving nonlinear multiobjective problem can be described in the following steps.

Step (1):

Solve the i[th] sub problem to obtain $(\overline{M}^{o}_{i}, f^{o}_{i})$:

Minimize $f_i(x)$, $i = 1, 2, \ldots\ldots, r$
Subject to
$x \in \overline{M}$

Where:

$\overline{M}^{o}_{i}$ is the set of optimal solution of the i[th] objective.

f^{o}_{i} is the optimal value of the i[th] sub-problem.

Step (2):

Using the ε-constraint technique for solving the nonlinear multiobjective programming problem to obtain all the weakly efficient solutions with respect to optimizing one of the objectives (for example f_w). Let ε_i is placed by $f^{o}_{i} + \eta_i$ in case of minimization and by $f^{o}_{i} - \eta_i$ in case of maximization where $i \neq w$ and $i = 1, 2, \ldots, r$. The set of the weakly efficient solutions is denoted by $\overline{M}^{o}_{\eta w}$, for the objective f_w.

The balanced number η_i represents a measure of quitclaiming of f_i to reach its optimal value f^{o}_{i}.

$$
\begin{aligned}
&\text{Minimize } f_i(x) \\
&\text{Subject to} \\
&\bar{M}_{\eta_i}(\eta'_j) = \left\{ x \in R^n \left| \begin{array}{l} f_i(x) - f_i^o \le \eta_i \\ g_j(x) \le \eta'_j \end{array} \right. \right\}
\end{aligned}
\tag{2}
$$

Where $f_i(x)$ and $g_j(x)$ are convex functions, possessing continuous first order partial derivatives on the n-dimensional vector space R^n and η'_j are arbitrary real numbers.

If $\eta_i = 0$ for each i then problem (1) is balanced (objective functions are reconcilable), otherwise the problem is unbalanced (the objective functions are non-reconcilable).

Problem (2) is assumed to be stable for all $\eta'_j \in R^n$. For problem (2), the basic notions of parametric multiobjective programming problem that are considered in [1, 3] will be defined and will be analyzed qualitatively for parametric non reconcilable multiobjective programming problem. This will be done by using a balanced number η.

Definition 2.1 "feasible parameters with a balanced number η"

SF_η is the set of feasible parameters with a balanced number η of problem (2), which was firstly introduced by Osman [6] is

$$SF_\eta = \left\{ \eta \in R^r \left| \bigcap_{i=1}^{r} \bar{M}_{\eta_i}(\eta'_j) \neq \phi \right. \right\}$$

Remark:

The set of feasible parameters with a balanced number η, is nonempty since $0 \in SF_\eta$.

Definition 2.2 "solvability set with a balanced number η"

SV_η is the solvability set with a balanced number η of problem (2) is defined by:

$$SV_\eta = \left\{ \eta' \mid m_{opt}(\eta') \neq \phi, \eta' \ge 0 \right\}$$

Minimize $f_i(x)$, $\quad i \in \{1, 2, ..., r\}$

Subject to

$x \in \bar{M}$

These sub problems are solved separately to obtain the set of optimal solutions $\bar{M}_i^o$ and its corresponding optimal value f_i^o for the i[th] objective where:

$$\bar{M}_i^o = \{x \in \bar{M} \mid f_i(x) = f_i^o\}$$

In the proposed algorithm, the ε-constraint technique for solving the multiobjective programming problem is used in a modified form. The ε-constraint technique has the following Characteristics [4]:

- The solution of ε-constraint problem is weakly efficient solution.
- A point $x^* \in \bar{M}$ is efficient solution if it is a unique solution of ε-constraint problem with $\varepsilon_j = f_j(x^*)$

2. Qualitative Analysis of Basic Notions

Parametric programming is a natural extension to mathematical programming, dealing with the effects of variations of the parameters on the obtained optimal solutions, (see [1], [3], [5]). This concept can be applied in the same way to non-reconcilable multiobjective nonlinear programming problems. The number η_i is considered to be the principle key in determining the best efficient solution and leads to the concept of:

- The set of feasible parameter with a balanced number η which is denoted by SF_η
- The solvability set with a balanced number η which is denoted by SV_η
- The stability set of the first kind with a balanced number η which is denoted by $S_\eta^{(1)}$

Consider the i[th] objective problem in a vector convex programming problem with a balanced number η of problem (1). It has the following form:

1. Introduction

The proposed approach is a recent technique to obtain the best efficient solution, in case of having no priority among the objectives. A distinctive feature of this approach is that the ability to find the best efficient solution among the efficient solutions according to a new measurement which is called η-measurement principle.

Mathematically, the multiobjective optimization problem can be formulated as a vector optimization programming problem under certain constraints [7]. One of the well known techniques for solving the multiobjective programming problem, the ε-constraint method is a procedure which overcomes some of the convexity problems of the weighted sum technique. It involves minimizing (in case of minimization) a primary objective and expressing the other objectives as constraints. This method is able to identify many efficient solutions on a non-convex boundary, which are not obtainable using the weighted sum approach [2].

In this paper, we shall solve a multiobjective programming problem by introducing the concepts of a balanced number η and η-measurement principle. The balanced number η_i represents a measure of quitclaiming of the ith objective f_i to reach its optimal value f_i^o. In section 2, the qualitative analysis of basic notions is introduced. The algorithm for solving the problem is given in section 3. Section 4 introduced illustrate examples to clarify the idea.

1.1.Problem Definition

Consider r-vector function defined on a compact set $\bar{M}$, of the following form:

$$\begin{aligned} &\text{Minimize } F(x)=\{f_1(x),\ldots\ldots\ldots,f_r(x)\} \\ &\text{Subject to} \\ &x \in \bar{M} \end{aligned} \qquad (1)$$

Where:

$$\bar{M} = \left\{x \in R^n \mid \; g_j(x) \le 0, j = 1,\ldots\ldots, m\right\}$$

The components of $F(x)$ defined separate optimization problems of the form:

η-Measurement for Finding the Best Efficient Solution for Multiobjective Programming Problems

Bayoumi M. Ali[1]

Abd El-azeem Mohammed[2]

Abstract:

A technique for finding the best efficient solution among the efficient solutions for multiobjective programming problem is presented. This approach introduces the concept of a balanced number η for the multiobjective programming problem and η-measurement principle for finding the best efficient. A parametric study of the set of parameters is given in terms of the balanced number η.

Key words:

Multiobjective Programming – Balanced Number Concept – The η-measurement principle.

[1] Associate Professor – Cairo University – Faculty of Computer Science and Information – DS Dep.

[2] Cairo University – Institute of Statistical Studies and Research – OR Dep.

References

Artana, K. B. and K. Ishida (2002). "Spreadsheet modeling of optimal maintenance schedule for components in wear-out phase." Reliability Engineering & System Safety 77(1): 81.

Baliwangi, L., H. Arima, K. Ishida and K. B. Artana (2006). Optimizing ship machinery maintenance scheduling through risk analysis and life cycle cost analysis, Hamburg, Germany, American Society of Mechanical Engineers, New York, NY 10016-5990, United States.

Clark, G., G. C. P. Mehta and T. Thomson (1992). Application of knowledge-based systems to optimised building maintenance management. Proceeding of the Fifth International Confrence on Industrial and Engineering Applications of Artificial Intellegence and Expert Systems. Paderborn, Germany.

Deris, S., S. Omatu, H. Ohta, S. Kutar and P. A. Samat (1999). "Ship maintenance scheduling by genetic algorithm and constraint-based reasoning." European Journal of Operational Research 112(3): 489.

Elkhouly, N. (2001). Maintenence Planning, College of Engineering Studies and Technology.

Graham, J. H., G. Brockman and J. J. Shea (1995). "Hybrid Expert Maintenance Advisory System for Naval Systems Applications." Naval Engineers Journal 107(1): 23.

KOTC. (2009). "Kuwait Oil Tanker Company S.A.K." Retrieved 01/03/2009, 2009, from http://www.kotc.com.kw/.

Kralj, B. and R. Petrovic (1995). "A multiobjective optimization approach to thermal generating units maintenance scheduling." European Journal of Operational Research 84(2): 481.

Oke, S. A. (2006). "An approach for evaluating preventive maintenance scheduling cost." The International Journal of Quality & Reliability Management 23(6-7): 847.

Schrage, L. (1999). Optimization Modeling With Lingo Lindo Systems.

Smits, S. and D. Pracht (1991). Moca--A Knowledge-Based System for Airline Maintenance Scheduling. Proceedings of the The Third Conference on Innovative Applications of Artificial Intelligence, Association for the Advancement of Artificial Intelligence Press.

6 Conclusion

In this paper a maintenance scheduling methodology is proposed to optimize the scheduling of dry-docking for maintenance of the vessels in the KOTC fleet. The basic constructs of the model illustrate a zero-one integer programming problem; mixed integer programming techniques have been shown to be a useful approach to scheduling maintenance tasks. The implementation of the model proves the applicability of the method to the scheduling of a merchant shipping company's dry-docked maintenance activities.

The objective function of the model was to maximize ships' availability. The objective function can be extended to include the cost of the dry-docking maintenance activities. In further research, it is anticipated that multiple constraints could be added to the model to accommodate the budget for the dry-docking and the demand for the ships in different seasons. Fuzzy parameters could also be applied to cover any variation in the input data. A stochastic approach may also yield greater power in detecting the variability in certain constraints.

The intention of the model is to help decision makers in planning and scheduling maintenance work; it could also aid operation researchers in understanding the relationships between the different processes.

carry out dry-docking maintenance is two. Therefore, the model used in the example offers successful results, as shown in Figure 3.

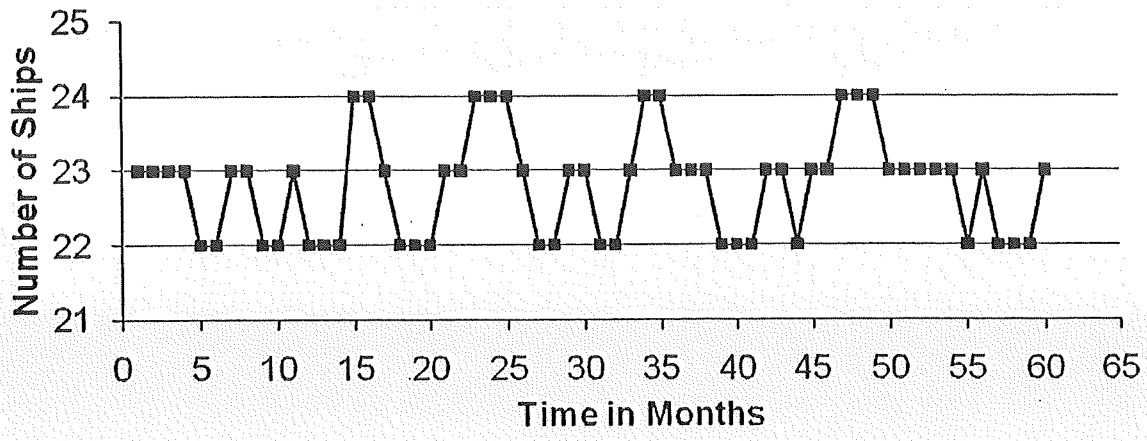


Figure 3: Ships' availability

From Figure 3 it can be seen that the model has maximized the ships' availability: 16.67% of the window period shows all 100% of the ships are available; 46.67% of the window period shows that only one ship is out of service for maintenance, which represents 96% of the ships as being in service; and 36.67% of the window period shows that two ships have been removed for maintenance, which represents 92% of the ships as being in service.

Type	Class	Ship	Period
T1	C1	S1	5-7
T1	C1	S2	36-38
T1	C1	S3	50-52
T1	C1	S4	44-46
T1	C2	S1	20-22
T1	C2	S2	53-55
T1	C2	S3	39-41
T1	C2	S4	12-14
T2	C1	S1	55-57
T2	C1	S2	42-44
T2	C1	S3	29-31
T2	C1	S4	24-26
T2	C2	S1	58-60
T2	C2	S2	9-11
T2	C2	S3	39-41
T2	C2	S4	17-19
T3	C1	S1	4-6
T3	C1	S2	1-3
T3	C1	S3	12-14
T3	C1	S4	31-33
T3	C2	S1	8-10
T3	C2	S2	27-29
T3	C2	S3	57-59
T3	C2	S4	18-20

Table 1: KOTC fleet dry-docking maintenance scheduling

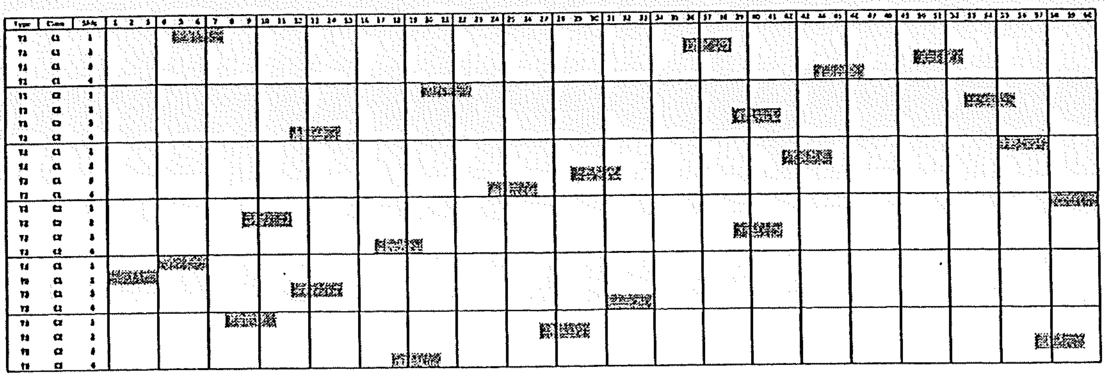


Figure 2: KOTC fleet dry-docking maintenance scheduling

The results from the example show that dry-docking maintenance is scheduled between months 1 and 60. The example also shows that once the dry-docking maintenance task starts, it will continue until it completes its duration, which in this case is three months. The example shows that the maximum number of ships scheduled for the shipyard to

5 Implementation and results

In order to validate the methodology, a scheduling of dry-docking maintenance for Kuwait Oil Tanker Company (KOTC) ships has been produced. Using data obtained from the KOTC fleet, which contains 24 vessels and is comprised of three different types of tanker (t=1 for crude oil tankers, t=2 for product oil tankers, and t=3 for gas tankers), each type having two different classes and a planning horizon of 60 months.

The maintenance constraints are the maintenance window, maintenance completion, and ship limits. For the maintenance window, the earliest and latest maintenance limits can be determined from the previous experience of ships' engineers (this information is based on expert judgement rather than a mathematical approach); the maintenance period usually starts at the earliest chosen time, which is month one $E_{tcs} = 1$, and the end of latest chosen time, which is month 60, $L_{tcs} = 60$. The maintenance period on average takes from two to three months; in this example the maintenance period chosen is set to three months. The completion constraint is chosen to ensure that if a ship is in dry dock it has to complete the docking time without any interruption to the work. The ship limit constraint ensures that the minimum number of ships is in the dockyard for maintenance during the same period. In the example, the maximum number of ships in the fleet allowed to be in dry dock is two. An integer programming approach has been used to solve the model by using the LINGO modelling and optimization package (Schrage 1999). The results are demonstrated in the table and figure shown below, which identifies the dry-docking period for every ship in the KOTC fleet.

4.2.3 Ship limit constraints

There are always a maximum number of ships which can go for maintenance during certain periods without the shipping company experiencing an unacceptable loss of fleet carrying capacity or revenue earning potential. A limited number of ships of one type in each class can, therefore, be sent for maintenance work and the remaining ships must stay in operation. Therefore, this constraint is used to limit the number of ships of one type in one class which can be sent for maintenance at any one time. The mathematical representation of such a constraint can be presented as follows:

$$\sum_{c=1}^{n} x_{tcsk} \geq n - r \qquad (5)$$

Where k runs across all time intervals: $k = 1,2,3,\ldots..,L$ and r is input data which indicate the maximum number of ships that are allowed to be maintained in period k.

4.2.4 Objective function

The objective function of the model can be presented as follows:

$$\text{Max.} \sum_{t=1}^{T} \sum_{c=1}^{m} \sum_{s=1}^{n} \sum_{k=1}^{L} x_{tcsk} \qquad (6)$$

Where x_{tcsk} represents the number of ships s available in class c of type t throughout the maintenance planning period k.

Subject to:

$$x_{tcsk} \leq y_{tcsk} \qquad (7)$$

And equations (1) – (5)

To model this constraint, a zero-one decision variable should be introduced which will represent the start of a ship's maintenance period.

Therefore, let

$$S_{tcsk} = \begin{cases} 0 \\ 1 \end{cases}$$

Where 0 if ships of the class c in type t starts it's maintenance on period k, otherwise 1

Therefore, the maintenance completion constraint will take the following form:

$$y_{tcsk} = 1 - k + \sum_{q=1}^{k} S_{tcsq} \qquad (1)$$

$$\text{for} \quad E_{tcs} \le k \le E_{tcs} + D_{tcs} - 1$$

$$y_{tcsk} = 1 - D_{tcs} + \sum_{q=k+1-D_{tcs}}^{k} S_{csq} \qquad (2)$$

$$\text{for} \quad E_{tcs} + D_{tcs} \le k \le B_{tcs}$$

$$\sum_{k=1}^{B_{tcs}-D_{tcs}+1} S_{tcsk} = B_{tcs} - D_{tcs} \qquad (3)$$

$$\sum_{k=1}^{B_{tcs}} S_{tcsk} = B_{tcs} - E_{tcs} \qquad (4)$$

Where D_{tcs} = duration of maintenance for ship s in class c of type t.

The first equation ensures that a job will be completed once work begins on any ship s in period k. The other two equations ensure that in period ($B_{tcs} - D_{tcs}$) and onwards, no new maintenance job will be started but that in these periods the maintenance jobs which have been started can be completed.

4.2.1 Maintenance window

According to the requirements of the classification society, a ship must go into dry dock for maintenance regularly (every five years for a major survey and key maintenance) in order to keep the ship's efficiency at a standard level. This can be achieved by specifying the latest time that the ship can be operating without maintenance and the earliest time it can be placed under maintenance. Mathematically the maintenance window can be expressed as follows:

$$y_{tcsk} = \begin{cases} 1 & \text{if } k < E_{tcsk} \text{ or } k > B_{tcs} \\ 0 & \text{if } E_{tcs} \le k \le B_{tcs} \end{cases}$$

Where:

E_{tcs} = Earliest time that ship s of the class c of type t can be taken for maintenance

B_{tcs} = Latest time that ship s of the class c of type t can be taken for maintenance

Thus y_{tcsk} is fixed at 1 before the earliest and after the latest times to allow for the starting period for maintenance of ship s in class c of type t and can be 0 or 1 between those times.

4.2.2 Maintenance completion

The purpose of this constraint is to ensure that the maintenance time (dry-docking period) for each ship occupies the required duration without interruption. This means that once the ship is in dry dock for maintenance, the work must be carried out without stopping until the maintenance is finished and the ship returns to operation.

where:

t represents the type $t = 1,2,3,\ldots\ldots,T$

c represents the class $c = 1,2,3,\ldots\ldots,m$

s represents the ship $s = 1,2,3,\ldots\ldots,n$

k represents the number of the planning horizon period k=1,2,3,……,L

The decision variable x_{tcsk} is set to 0 for two reasons:

1. When the ship s in class c of type t is undergoing maintenance work during period k, ($y_{tcsk} = 0$).

2. When it is idle, ($y_{tcsk} = 1$).

Thus, when the ship is not under maintenance ($y_{tcsk} = 1$), this does not necessarily imply that it is in operation ($x_{tcsk} = 1$), as it could simply be idle. Therefore, the following constraints are needed to link variables x_{tcsk} with variables y_{tcsk}.

$x_{tcsk} \le y_{tcsk}$ for all $t,c,s,$ and k.

4.2 The set of constraints

The ship maintenance scheduling problem is one of constraints optimization. The objective function has to be maximized or minimized according to certain constraints. In this model the following constraints will be considered:

1. Maintenance window
2. Maintenance completion
3. Ship limit constraints

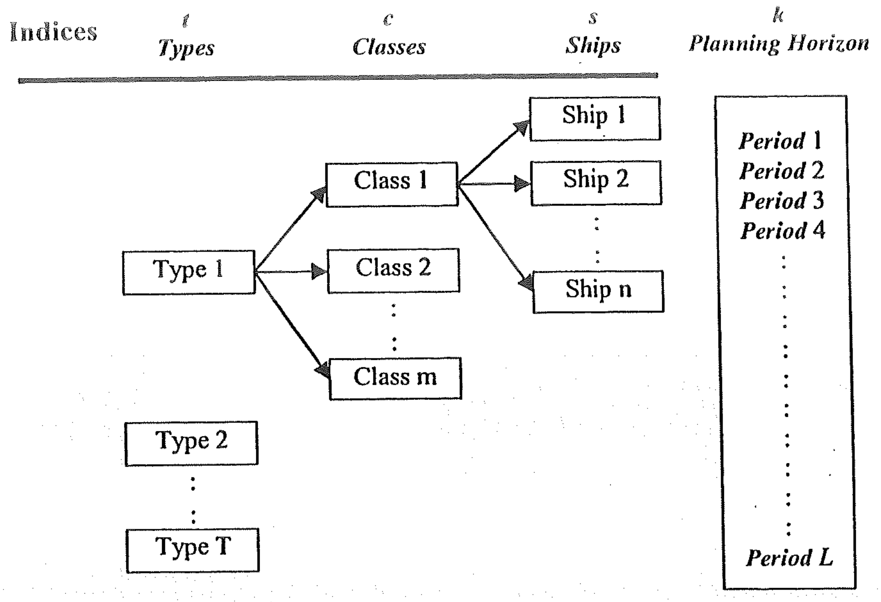


Figure 1: Model indices

4.1 Decision variables

The first step in constructing the mathematical model is to identify the decision variables which may be controlled and serve to determine the outcome of the maximization or minimization decisions. The development of such values will provide the optimal solution. For dry-docking ship maintenance scheduling problems the decision variables are designed as follows:

$$x_{tcsk} = \begin{cases} 1 \\ 0 \end{cases}$$

Where 1 is the ship s of the class c in type t during the period k in operational status, otherwise, 0

Also, define

$$y_{tcsk} = \begin{cases} 1 \\ 0 \end{cases}$$

Where 1 if ship s of the class c in type t is not in maintenance during period k, otherwise 0

4 Mathematical model formulation

The mathematical model of ship maintenance scheduling is presented in this section, using the zero-one integer-programming.

The first step after identifying the problem in model formulation or construction is to establish the decision variables, which are the controllable parameters whose values can be controlled by the decision maker and which affect the functioning of the system, in this case zero-one values. An objective function is then identified which should satisfy all constraints on the decision variables. The objective function for this model is to maximize the availability of the ships. The constraints to which the model is subjected have limited values.

A shipping fleet typically contains various different types of vessels, with each type comprised of several classes of ships of that type, and the fleet maintenance planning horizon is usually around 5 years or 60 months. The aim is to schedule the dry-docking maintenance tasks for different types of vessels (t=1,.....,T) each type consisting of different classes (c=1,.....,m) and ships (s=1,........,n) in order to maximize each ship's availability over the planning period (k) subject to ship maintenance constraints. Figure 1 shows the indices used in this model.

3 Description of the problem

In any merchant shipping company dry-docking maintenance is very important for ensuring that ships are seaworthy and ready to carry out the operations required. This can only be guaranteed if effective planning and management of maintenance is achieved.

In the current situation, dry-docking maintenance is carried out based on two factors; previous dry-docked maintenance and the due date for the classification society survey. However, it is clear that other factors must also be considered in order to optimize the dry-docking maintenance of ships; this will include factors such as the demands on a ship's operations as well as dockyard availability.

As emphasized previously, the availability of vessels is paramount to commercial shipping companies; the corporate risk management strategy is based on this availability and therefore, unanticipated or long duration maintenance operations are likely to have a significant impact upon profitability and revenue. Therefore, dry-docking maintenance should be scheduled in an optimum way; this could be achieved by applying a mathematical approach to identifying the best possible solution to scheduling. This paper therefore adopts a numerical solution to the problem using a zero-one integer linear programming model.

Artana and Ishida (2002) address a method for determining the optimum maintenance schedule for components in the wear-out phase. A case study and sensitivity analysis on a liquid ring primer in a ship's bilge system were used. The interval between maintenance for the components was optimized by minimizing the total cost. A premium solver platform, a spreadsheet-modelling tool, was utilized to model the optimization problem. The optimization process investigates n-equally spaced maintenance at an interval of Tr.

Baliwangi et. al. (2006) developed a methodology based on the integration of ship maintenance scheduling management with risk evaluation in a Life Cycle Cost (LCC) approach. The methodology was designed to establish the optimal maintenance scheduling in several steps, which include determining component function, generating the predicted timings and possible component combinations, analysing the associated alternatives and uncertainties, and lastly selecting the best alternative using a criterion LCC.

Oke (2006) have redefined the expression that defines the period-dependent cost function for preventive maintenance scheduling activity. The approach involves transforming the preventive maintenance cost function that is expressed in terms of several variables into a more precise framework. A case study from the shipping industry is also presented.

In other industries, Kralj and Petrovic (1995) have studied the maintenance scheduling in power generation. Clark et al. (1992) studied the maintenance scheduling of buildings and Smits and Pracht (1991) have studied the maintenance scheduling of an airline.

scheduling problem. A branch and bound algorithm is used to solve the integer programming model.

An introduction to the dry-docking maintenance of ships has been presented in Section One. This is followed by a brief literature review of maintenance scheduling in Section Two. Section Three gives a description of the main problems associated with the scheduling of dry-docked maintenance. The mathematical model formulation is discussed in Section Four. The production of a recommended schedule for dry-docking maintenance for the Kuwait Oil Tanker Company fleet is presented in Section Five to illustrate an application of the model. The paper concludes with a summary and draws attention to the benefits of the proposed model.

2 Literature review

There has been little research carried out in ship maintenance scheduling; most of the studies focus on maintenance repair diagnosis (Graham et al. 1995). The most widely-used techniques in maintenance scheduling are knowledge-based systems, mathematical programming, and the branch and bound algorithm.

Deris et. al. (1999) model ship maintenance scheduling as a constraint satisfaction problem (CSP) to maximize the availability of either a ship, a squadron or a fleet for operations which have satisfied maintenance requirements, dockyard availability and operational requirements. The variables used in the model are based on the start times and their domain values are the start and horizon of the schedule. An application of this model was used by the Royal Malaysian Navy. Constraint-based reasoning (CBR) was adopted, which requires start times for the first activities of maintenance cycles. Genetic algorithm (GA) is then adopted to establish the start times of the first activity.

1 Introduction

Efficient scheduling of ship maintenance requiring dry-docking is a vital component in the overall strategic management of a merchant shipping fleet; where this process is managed well, significant revenue and efficiency gains can be made since downtime is in theory minimized. Ordinarily ships are usually scheduled to undergo maintenance at a shipyard every two-and-a-half years for an intermediate classification survey and every five years for a major one (Elkhouly 2001). Shipping companies ordinarily attempt to carry out only the five-yearly major surveys with the classification societies; this is particularly the case with new vessels. This is possible due to the specification to which new ships have been built in recent years being to a much higher standard of construction than was previously the case. For example, the Kuwait Oil Tanker Company (KOTC) has proved to the classification society that ships built since the 2007's only need undergo five-yearly dry-docking (KOTC 2009).

In dry-docking, a ship is removed from the water to enable work to be performed on the exterior part of the vessel, which is ordinarily below the waterline. The owners usually plan and schedule their ships for dry-docking based on the dates of previous maintenance and on the latest date by which the classification society's inspection requirements must be met. The problem is that the scheduling is not planned to provide the optimum solution; therefore, optimum scheduling is needed to maximize the availability of ships. In this paper an optimum scheduling solution for the five-year major surveys is presented by a mathematical formulation which has been developed for the scheduling of the dry-docking maintenance of ships. The mathematical technique used is an integer model in order to formulate the relationship between the variables and solve the resultant

Dry docked maintenance scheduling of commercial shipping vessels in Kuwait using an integer programming methodology

Yousef Alhouli [1], David Ling [1], Richard Kirkham [1], Taha Elhag [2]

[1] School of Mechanical, Aerospace and Civil Engineering, University of Manchester, Manchester, M60 1QD, UK
[2] The Bartlett School of Construction and Project Management, University College London, London, UK

Abstract

The scheduling of ship maintenance that requires the use of a dry dock is an important aspect of the management of a merchant shipping company's fleet. The decision to remove a vessel from operational service in order to facilitate dry-docked maintenance activities will naturally impact upon overall availability levels of the fleet (as well as flexibility and downtime), which in turn can have a bearing upon corporate revenue and cash-flow levels.

Operators therefore seek to manage the scheduling of these maintenance activities in an optimal way, using a combination of operational research based techniques and strategic management decision-making. Effective scheduling has an implied benefit to the dockyard operators in facilitating forward-planning of essential ship's maintenance tasks, which should ensure that the supply-chain is adequately prepared. This joined-up approach to scheduling will also provide useful data to plan and schedule the routing and docking of ships en-route to the dockyard.

The research described in this paper demonstrates the use of an integer linear programming model to facilitate the scheduling of dry-docked maintenance activities in an optimum way designed to maximize the availability of ships within the fleet. The optimization model is defined by three constraints; maintenance window, maintenance completion, and ship limit. The model used data from the Kuwait Oil Tanker Company's fleet which has 24 tankers of three different types. The results of this modelling approach demonstrate that the availability of the company's ships over a time horizon of 60 months is maximized and therefore potentially optimal in terms of impact upon corporate revenue (ceteris paribus). The limitations of this model are discussed and further work required to refine this approach in an operational setting is discussed.

Keywords: mercantile operations management, mercantile maintenance, operations scheduling, integer programming

Index

Operations Research

Cairo University

Institute of Statistical Studies and Research

THE ANNUAL CONFERENCE
ON STATISTICS,COMPUTER SCIENCE
AND OPERATIONS RESEARCH

Operations Research

7-10, Dec ,2009