Optimization of a Hospital Distribution System:
The Case of Central Warehouse of Hospital de Santa Maria

Diana Buzaglo

Department of Engineering and Management, Instituto Superior Técnico, Technical University of Lisbon, Portugal

Abstract

This work aims at the minimization of the makespan in an identical and parallel machine problem, without preemption, through the scheduling of jobs with variable process duration which depends on its location in the processing sequence. Job durations are known along the time horizon.

The work's case study is based on the distribution system of clinical material, previously placed on the picking car for transportation by the picking team. The distribution starts at the Distribution Room of the Central Warehouse of the Santa Maria's Hospital and ends at each hospital service. Deliveries are done every working day, from 9 a.m. to 5 p.m., following a pre-made list of services to satisfy. One job corresponds to one delivery to a service, including the round trip of the transportation of the picking car from the Distribution Room to the service located anywhere in the Hospital.

Through a heuristic model fitting the case study's problem, one proposes a job sequence knowing that the availability of lifters is linked to the frequency with which the lifters are used throughout the day. The problem is optimized through a mixed integer linear programming (MILP) model that minimizes the makespan of the complete distribution list per day. The mathematical model was implemented in the modelling system GAMS and solved using CPLEX.

Keywords: Makespan, Parallel Machine, Short-Term Scheduling without preemption, Manual Distribution of Clinical Material in a Hospital.

1. Introduction

The presence of clinical material in each clinical service in any hospital is a critical requirement since health care service could not be brought to people without these materials. Therefore the distribution system of clinical material in a hospital is one of the most important variables which dictate the performance of the hospital service.

The distribution system is based on the delivery and storing of items in each clinical service with origin in a central warehouse. Considering that the distribution is performed by a team of collaborators and each collaborator distributes to a set of different services, each clinical service is satisfied in a sequential way. Thus, the problem is similar to a parallel machine short-term scheduling (without preemption) problem, where each member of the distribution team and delivery activity is a machine and a task, respectively. The main objective in this work is to minimize the makespan of all tasks scheduled for each work day of the week considering the number of machines available.

In the scientific community the short-term scheduling and the minimization of the makespan is widely studied. Taking into account the similarities between the case study and the production scheduling area, the research was based in papers in this topic. Health services scheduling literature research was also done, but no similar problems were found.

Pinedo (2002) says that different tasks scheduling have different impacts on the performance of the system in question. So it is worth spending resources to find the best task schedule as an alternative to choose a random task schedule. But its conceptualization and implementation, depending on the quality required of the solution, is difficult in a technical point of view.

The task scheduling is the activity with major importance at the operational level. This kind of activity is defined by the allocation of resources (equipment, manpower or space) to different kinds of activities within a given time horizon (Mariano, 2008). Task scheduling is always necessary when there is a competition for the use of the same resources between different activities (or tasks), since those resources are limited (Reklaitis, 1992).

In the parallel machine problem, each machine is one of a set of that type of resource. The one machine problem consists in determining the processing sequence of all tasks, knowing that each task must be processed one time exactly, processing times are previously known and the machine cannot be inactive whenever there are still tasks to be processed (Pannerselvam, 2006).

If there are different kinds of processes for each task, this is a two or more non parallel machines problem, as in flow shop, job shop and open shop models.

The parallel machines problem is another possible scheduling problem. It is a generalization of the one machine problem or of the flow shop model. By definition, in the parallel machines problem the machines are identical which means that each machine has equal processing times, and each task must be processed by one of the machines. Machines are not allowed to be inactive when there are tasks to be processed (Pinedo 2002).

For the purposes of our study, the parallel and identical machine problem (without preemption) takes into account some assumptions (French, 1982):

- Each task has to be processed one time exactly and cannot be processed simultaneously by more than one machine;

- Each machine can only process one task at a time;

- All tasks can be processed by any machines;

- The possibility of a broken machine is not considered, which means that all machines are always available;

- It does not exist a setup time for the machines;

- The number of machines and tasks is previously known and it is fixed;

- The processing time of each task is previously known and equal for any machine.

The minimization of the makespan is the common objective for this type of problem since usually it is required to minimize the global needed time to process all tasks. Hence the value of the makespan depends on the processing sequence of tasks for each machine (Pinedo, 2002).

The mathematical model for an identical and parallel machine with the minimization of the makespan, as the objective function, is presented by Pinedo (2002). Another model, given by Hashemian (2010), is studied. This one considers the possibility of a machine being unavailable during a period of time. Other mathematical representation methodologies can be used, such as the state-task network (Kondili et. al, 1993) and the resource-task network (Pantelides, 1994).

In terms of solution method, the branch-and-bound, an exact method, gives the optimal solution, but its computational process can be quite large (Hillier e Lieberman, 2005). Alternatively, there are meta-heuristics, as the tabu search, the simulated annealing and the genetic algorithm, that have a faster solving process but the given solution may not be the optimal (Hillier e Lieberman, 2005). The simulation is another technique widely used in the area of operational research (its definition is by Hillier e Lieberman (2005) and Cheema (2005)). Cheng and Sin (1990) compare the absolute performance ratio of others algorithms that are more used in the resolution of parallel machines, without preemption, searching for the minimization of the makespan. These algorithms are based on the construction of a priority list (or scheduling list) of tasks to be allocated to the machines. Graham (1966, 1969) proposed the random algorithm, in which the list is made randomly. After that he proposed a better algorithm, the pairwise interchange (Graham et al., 1979). Later, another random algorithm was presented by Achugbue and Chin (1981) with a better ratio. Another kind of algorithm, the Longest Processing Time, was studied by De and Morton (1980). Later, in the text of Ibarra e Kim (1976), the same algorithm is given but improved. In the text of Graham (1969), he also proposes the K algorithm. After that, a better algorithm is given by Sahni (1976 and 1977), the rounded dynamic programming algorithm.

In this paper, the scheduling of tasks in an identical and parallel machine (without preemption) problem, with minimization of the makespan, is applied to a real case of a Portuguese hospital in Lisbon, the Hospital of Santa Maria (HSM). The purpose of this study is to improve the work done by its distribution team by scheduling the delivery tasks, knowing previously its processing times during the work hours, between 9 a.m. and 5 p.m., and the list of which services are to be satisfied (tasks). Tasks have variable process duration which depends on its location in the processing sequence, considering a discrete time horizon.

2. Case Study

The HSM is one of the largest hospitals in Lisbon. Its clinical services are distributed all over the 9 floors of the building, and each floor has an area of about 60 x 200 square metres.

In the year 2006 there were some reforms made in the logistic system of the HSM. Since then all the clinical material is stored in the Central Warehouse (CW) of the HSM. It was also implemented a computer application in order to register all kind of movement of each item stored in the CW (consumption, requests, orders, items picket and received in the CW). This application allowed the elimination of the use of paper for requests made by clinical services. The implementation of the Kanban system was one of the main improvements. Since all the improvements took place, the logistic activities proved much better than before (according to the Director of the Logistic Department).

The Kanban system is an automated request made immediately after the consumption of the item is registered by its service. There are 145 clinical services in HSM, 104 of which use this system and have its own peripheral warehouse (PW) (named as the Kanban service).

The distribution team is responsible for the delivery of the items requested by the Kanban services. Each day of work there are pre-made lists of Kanban services that are satisfied. The items to be delivered to a Kanban service are transported by a picking car, which is previously prepared by the picking team of the CW. Note that if the items of two or more Kanban services fit in the same picking car and if those services are locally near, the distribution team member needs only to transport one picking car.

Each task consists in bringing the filled picking car from a location near the Distribution Room to the Kanban service(s), using the lifter. It is considered that there are 12 lifters available for the transportation of the picking car between floors. When the distributor arrives at the respective PW, he has to replace all the items in its designed storage location. After finishing this, he returns to the initial location using the safe lifter and makes a 10 minutes break. Every time that a lifter is used, the user has to wait its availability (which requires enough space for the picking car).

The duration of a task depends on the time of the day that the task takes place. The delivery is made through the use of lifters. Since their availability is linked to the number of people present in the hospital, the same task can be processed in a longer or shorter time.

Between all teams of the CW (reception, picking and ordering management) the distribution team was the only team that did not report any improvement brought by the computer application. It seems that the team presents an effective work but not efficient, since it does all the deliveries from the pre made list but it takes too long to do it.

3. Problem Characterization

Knowing that processing time of each task is different given the time of day of its execution, a data collection was made, concerning the time needed for the transportation of the picking car from all 12 lifters during the working time of the distribution team (the lunch hour was not included). Through this data collection, one was able to identify the phases of the day for which the transportation time was relatively constant (no more than 5 minutes of variation).

Another data collection was made concerning the time needed for the replacement of all items in each PW.

Therefore the processing time of a task includes the transportation time, replacement time and break time.

In determining the tasks processing time the following assumptions were considered:

- The number of tasks per work day, its processing time and the number of distribution team members (machines) are previously known;

- Each task has to be processed one time exactly by any machine. All machines are identical, thus the processing time is the same for every machine.

- Each machine cannot process simultaneously more than one task, and each task cannot be simultaneously processed by more than one machine;

- All task in the pre made list of deliveries for each work day, have to be executed within the working hours (9 a.m. to 5 p.m., except lunch hour between 1 p.m. and 2 p.m., however the beginning and ending of the lunch hour is variable).

- Once the task has begun, it cannot be stopped;

- The beginning and end of a task must be in the same phase of the day;

- In each task it is taken into account a 10 minute break for the machine;

- There are no priority tasks;

- The transportation of items is always via the picking car;

- It is possible to bring items of different Kanban services in the same picking car, if there is enough space inside the picking car and when the services are near from each other;

- A normal week is a week without holidays on working days, outside the holiday period and the quantity of items replaced is predictable (without peaks of consumption);

- The data collection was made during regular weeks. So the replacement time is constant. That means the transportation time is the responsible for the task processing time variation;

- Since the tasks beyond the AC (laundry, catering, trash disposal, consult and visit scheduling and shift scheduling) are performed in similar manner every working day, it means that the variability of task processing time is the same for each work day;

- The pre made list of which Kanban services are satisfied each work day is previously known. When there is a holiday, the list of services is made in advance, taking into account that the services scheduled for the holiday are satisfied in the day before and after the holiday;

- The data was collected during 33 work days in normal weeks. It proved that there are three phases in each day.

Since the researched studies didn't take into account the variability of the processing time of each task, a mathematical model was made from scratch.

4. Model Formulation

The proposed MILP model calculates the best task sequence for each machine minimizing the makespan. The outcome is a task schedule for each machine in each phase of the day. The nomenclature and formulation is presented bellow.

Indexes:

i – Tasks

m – Machines

t – Day phases (or phases)

p – Processing position of a task

Sets:

$I = \{1, 2, \dots, i, \dots, NI\}$ Set of tasks to be processed

$M = \{1, 2, \dots, m, \dots, NM\}$ Set of machines

$T = \{1, 2, \dots, t, \dots, NT\}$ Set of phases of the day

$P = \{1, 2, \dots, p, \dots, NP\}$ Set of sequential processing positions for each machine in each phase

Parameters:

ti_t – Start of phase *t* (minutes)

tf_t – End of phase *t* (minutes)

$tp_{i,t}$ – Processing time of task *i* in phase *t* (minutes)

$maxt = max\,\{tf_t : \forall\, t\, \in T\}$ – Maximum time allowed (minutes)

Variables:

Decision variable:

$y_{i,m,p,t}$ – Binary variable which is equal to 1 if task *i* is processed by the machine *m* in position *p* during the phase *t*

Positive variables:

$Tinic_{i,m,p,t}$ – Start time of task processing *i* by machine *m* in position *p* during the phase *t*

$Tproc_i$ – Processing time of task *i*

$Cmax$ – Makespan

Objective function:

Equations (1) and (2) describe, mathematically, the makespan. By imposing the makespan as larger than any of the final processing times of any task, in any machine and for any day phase, the required objective function will be of the type min-max, in order to minimize the maximum final processing time.

$$Cmax \geq Tinic_{i,m,p,t} + Tproc_i\,, \quad \forall\, i \in I, m \in M, p \in P, t \in T \tag{1}$$

$$min\ Cmax \tag{2}$$

Equation (3) guarantees that each task is processed exactly once.

$$\sum_{m=1}^{NM} \sum_{p=1}^{NP} \sum_{t=1}^{NT} y_{i,m,p,t} = 1, \quad \forall\, i \in I \tag{3}$$

Equation (4) ensures that in each position of a machine in a given phase, is assigned a maximum of one task (this constraint defines the non-overlapping task execution).

$$\sum_{i=1}^{NI} y_{i,m,p,t} \leq 1, \quad \forall\, m \in M, p \in P, t \in T \tag{4}$$

The next constraint (Equation 5) makes sure that the processing positions are assigned on each machine in a successive way. Initially there are *NP* possible positions for each machine in each phase. If they are not fully used, the unused positions are allocated at the end of the sequence.

$$\sum_{i=1}^{NI} y_{i,m,p+1,t} \leq \sum_{i=1}^{NI} y_{i,m,p,t}\,, \quad \forall\, m \in M, p \in P, t \in T \tag{5}$$

Equation (6) calculates the processing time of each task according to the phase of the day it runs.

$$Tproc_i = \sum_{m=1}^{NM} \sum_{p=1}^{NP} \sum_{t=1}^{NT} tp_{i,t} \times y_{i,m,p,t}\,, \quad \forall\, i \in I \tag{6}$$

Equation (7) relates the start time of a task that is processed in subsequent position (*p+1*) by the machine *m* after processing the task immediately preceding (p). If task *i* is the subsequent to task *j* in machine *m* in phase *t*, the third term on the right side of this restriction is cancelled and the constraint becomes active. So the task *i* starts only after the task *j* ends (in other words its beginning – 1st term in the equation – adding to its processing time – 2nd term in the equation). If both *i* and *j* tasks are not processed in position *p+1* and *p* by the machine *m* in phase *t*, the last term of Equation (7) becomes nonzero, allowing the constraint to become non active, which means that this constraint does not influence the values of start time of these tasks.

$$Tinic_{i,m,p+1,t} \geq Tinic_{j,m,p,t} + Tproc_j - \left(2 - y_{j,m,p,t} - y_{i,m,p+1,t}\right) \times maxt \,, \quad (7)$$
$$\forall\ (i,j) \in I, m \in M, p \in P, t \in T$$

Equation (8) guarantees that the start time of a task, if it is allocated to the phase *t*, is less than the end time of this phase.

$$Tinic_{i,m,p,t} \leq tf_t \times y_{i,m,p,t} \,, \quad (8)$$
$$\forall\ i \in I, m \in M, p \in P, t \in T$$

Equation (9) allows the start time of a task, if it is allocated to the phase *t*, to be greater than the start time of this phase.

$$Tinic_{i,m,p,t} \geq ti_t \times y_{i,m,p,t} \,, \quad (9)$$
$$\forall\ i \in I, m \in M, p \in P, t \in T$$

Equation (10) ensures that, if the task is allocated to phase *t*, its duration should be less or equal to the duration of this phase (i.e. together with the two previous restrictions (8) and (9), Equation (10) ensures that a task begins and ends in the same phase).

$$\sum_{i=1}^{NI} \sum_{p=1}^{NP} tp_{i,t} \times y_{i,m,t,p} \leq tf_t - ti_t \,, \quad (10)$$
$$\forall\ m \in M, t \in T$$

The model formulation is made through 4 sets, which makes the execution of this model inefficient when trying to analyze a problem with large dimensions, since there are *NI x NM x NT x NP* binary variables to determine. In order to reduce the number of possible solutions, merely the possible values that the makespan can take are used. The lower limit of the makespan is given by the Equation (11). This equation selects, for each task, the shortest processing time between different phases. Then, the selected times are summed and its total is divided by the total number of machines. The upper limit is defined by Equation (12), where the makespan does not exceed the maximum time allowed.

$$Cmax \geq \frac{\sum_{i=1}^{NI}\left(\min\{tp_{i,t} : t \in T\}\right)}{NM} \quad (11)$$

$$Cmax \leq maxt \quad (12)$$

5. Results

The collected data shows that there are 3 day phases for each day. The first one begins at 9 a.m. and ends at 13 p.m. The following one begins at 2 p.m. and ends at 3.50 p.m. The last one begins at 3.50 p.m. and ends at 5 p.m.

The model was implemented using the software application tool GAMS 23.5 and solved using CPLEX 12.2. This application ran on a PC HP Pavillon dv6000, with an Intel Core2Duo processor and a 1GigaByte RAM. As stopping criteria a relative gap less than 10% was considered.

In this work, five scenarios were studied. The first one is the general case for each work day of a normal week, considering 4 machines (actual number of distributors). The second one is if the number of machines is decreased to 3. The third scenario studies what happens to the task schedule of a normal week when there is a holiday. Since the transportation of picking cars need the use of lifters, the fourth scenario presents the possibility of a lifter malfunction. Finally, the last scenario studies a work day during a holiday period.

Since the sets of the model, for each scenario of the problem, have large dimensions, the *NP* set dimensions was modified to the smallest value possible, in order to decrease the number of total iterations (equals to *NI x NM x NP x NT*). This is done using the smallest task duration and the largest phase duration.

The results are summarised in Table 1.

Table 1 - Results for each possible existing scenario in HSM

Scenario	Scenario characteristics	*Makespan*		*Relative* gap/ Processing time (min)	Comments
1	General case *NM*=4 *NP*=7 *NT*=3	Monday - *NI*=29	15h46	7,80 %/12	All deliveries are executed within the working hours. Monday is the day with the longest makespan. The results show that the third phase (3.50 p.m. – 5 p.m.) is not used. Actually, the distribution team ends all the deliveries from the pre made schedule between 4 p.m. and 4.30 p.m.
		Tuesday - *NI*=24	14h49	Optimal solution/9	
		Wednesday - *NI*=22	14h31	Optimal solution/4	
		Thursday - *NI*=22	14h39	Optimal solution/6	
		Friday - *NI*=28	15h34	6,20 %/30	
	Lunch hour not fixed	-		-	Varying the beginning and the end of lunch hour did not show any reasonable improvements (about 1%).
2	Less number of machines *NM*=3 *NP*=7 *NT*=3	Monday - *NI*=29	After 17h	-	The deliveries on Monday and Friday are not accomplished when using 3 machines (less one than the general case), Therefore, decreasing the number of distributors is not a viable option.
		Tuesday - *NI*=24	16h25	0,520 %/30	
		Wednesday - *NI*=22	15h31	2,72 %/30	
		Thursday - *NI*=22	16h17	Optimal solution/12	
		Friday - *NI*=28	After 17h	-	
3	Holiday on Tuesday Its deliveries are made on Monday and Wednesday *NP*=7 *NT*=3	Monday *NI*=29, *NM*=4	16h56	8,60 %/60	Since the distribution team does not work on holidays, the pre made list of deliveries for the holiday is split between the day before and after the holiday. With 4 machines all tasks are finished just before the end of work hours (5 p.m.), which gives very little flexibility for unforeseen or eventual error on the calculation of processing times. The presence of 5 machines guarantees the conclusion of all deliveries.
		Wednesday *NI*=22, *NM*=4	16h45	7,12 %/30	
		Monday *NI*=29, *NM*=5	15h33	8,65 %/34	
		Wednesday *NI*=22, *NM*=5	15h40	9,30 %/33	

4	Malfunction in a lifter *NI*=29 *NM*=4 *NP*=7 *NT*=3	Monday	15h50	5,86 %/50	Comparing the makespan of the same day in the general case, the occurrence of a malfunction in a lifter, did not show any reasonable aggravation (about 1,6 %). A possible malfunction in one of the lifters has no influence on the makespan.
5	Holiday period (during August) *NM*=4 *NP*=10 *NT*=3	Monday *NI*=29, *NM*=4	14h27	Optimal solution/29	During a holiday period, according to the Director of the Department of Logistics, there are 50% less people present in HSM than in general case. The difference of the makespan between the general case and a period of holiday is 22,8%.
		Monday *NI*=29, *NM*=3	15h48	4,12 %/41	The possibility of decreasing the number of machine is viable, since the makespan does not use the third day phase to schedule the tasks of the work day with the most tasks.

6. Conclusions

Hospital of Santa Maria is one of the largest hospitals in Lisbon, Portugal. It has 145 clinical services (out of which 104 are Kanban services). The clinical material deliveries done to the hospital's own services are only done to the Kanban services. For each working day the services to which the material is delivered are previously stipulated.

The scheduling of the delivery tasks performed showed that the team is effective in performing its tasks but not efficient.

Taking into account that the duration of each task varies according to the hour of the day it is executed, it was decided to solve the problem of the distribution team as a problem of task scheduling in parallel machines (with no preemption) in order to minimize the daily makespan.

The results have shown that in a normal week all daily tasks are complete before 4 p.m., one hour earlier regarding the exit hour for the distributors and that the current distributor number on the team is adequate, except when there is a holiday. In such weeks the addition of one extra element in the team is advised in order to guarantee all the deliveries.

In holiday periods, the team can go without one less element, since the total processing times are much inferior to those on a normal week.

The demonstrated results are essentially useful on a qualitative point of view (guidelines), since the data collection cannot be considered sufficiently extensive and, as such, the makespan value may not portray reality to its full extent.

In the future, HSM will be adopting a fully dedicated system for monitoring the time in the distribution itinerary and material replacements using the PDT to read the respective bar codes. This will allow for a much more accurate definition of the several daily phases and the associated results, returned by the model, will be more reliable both qualitatively and quantitatively.

In this work there are some others suggestions beyond solving the task scheduling problem in order to improve the distribution work, for instance at the level of the picking process (for instance, the picking team itself organizes in the same car the products for services closer to each other, so that the distribution team doesn't take time for this process), of the PW or of the type of requisitions made by clinical services.

On the continuity of this work a study of the PW is suggested regarding the improvement of the replacement time for the products and the control of their consumption. Other studies can be done in terms of variability of task duration. The model can be revised so as to account for different situations, such as allowing for a task to start and end in different phases or to improve model formulation.

References

Achugbue, J. O., Chin, F. Y. (1981) *Bounds on schedules for independent tasks with similar execution time.* Journal of ACM, Vol. 28, pp. 81-99.

Cheema, D. S. (2005) *Operations Research.* First Edition. Laxmi Publications.

Cheng, T. C. E., Sin C. C. S. (1990) *A state-of-the-art review of parallel machine scheduling research.* North Holland : European Journal of Operacional Research, Vol. 47, pp. 271-292.

De, P., Morton, T. E. (1980) *Scheduling to minimum makespan on unequal parallel processors.* Decision Science, Vol. 11, pp. 586-602.

French, S. (1982) *Sequencing and Scheduling: An Introduction to the Mathematics of the Job Shop.* Chichester. : Ellis Horwood Ltd.

Graham, R. L. (1966) *Bounds on certain multiprocessing anomalies.* Bell System Technical Journal, Vol. 45, pp. 1563-1581.

Graham, R. L. (1969) *Bounds on multiprocessing timing anomalies.* SIAM Journal of Applied Mathematics, Vol. 17, pp. 416-429.

Graham, R. L., Lawler, E. L., Lenstra, J. K., Rinnooy Kan A. H. G. (1979) *Optimization and approximation in deterministic sequenciong and scheduling: A Survey.* Annals of Discrete Mathematics, Vol. 5, pp. 287-326.

Hiller. F. S., Lieberman, G. J. (2005) *Introduction to Operations Research.* Eighth Edition. McGraw-Hill International Edition.

Ibarra, O. H., Kim, C. E. (1976) *On two-processor scheduling of one or two unit time tasks with precedence constraints.* Journal of Cybernetics, Vol. 5, pp. 87-109.

Kondili, E., Pantelides, C. C., Sargent, R. W. H. (1993) *A General Algorithm for Short-Term Scheduling of Batch Operations - MILP Formulation.* An International Journal of Computer Applications in Chemical Enginnering, Vol. 17, No. 2, pp. 211-227.

Mariano, P. (2008) *Optimização do Escalonamento da Produção das Máquinas de Papel de uma Indústria Papeleira.* Projecto de Dissertação de Mestrado : Universidade Técnica de Lisboa, Lisboa (Portugal).

Panneerselvam, R. (2006) *Operations Research.* Second Edition. New Delhi : Prentice-Hall of India Private Limited.

Pantelides, C. C. (1994) *Unified Frameworks for Optimal Process Planning and Scheduling*. Proceedings of the Second conference on FOCAPO. pp. 253-274.

Pinedo, M. (2002) *Scheduling: Theory, Algorithms, and Systems.* Second Edition. New Jersey : Prentice Hall.

Reklaitis, G. V. (1992) *Overview of Scheduling and Planning of Batch Process Operations.* NATO Advanced Study Institute—Batch process systems engineering. Purdue University, Antalya (Turkey).

Sahni, S. (1976) *Algorithms for scheduling independent tasks.* Journal of ACM, Vol. 23, pp. 116-127.

Sahni, S. (1977) *General techniques for combinatorial approximation.* Operations Research, Vol. 25, pp. 920-927.