Linear Programming Versus Convex Quadratic Programming for the Module Allocation Problem

Sourour Elloumi[1]

April 13, 2005

Abstract

We consider the Module Allocation Problem with Non-Uniform communication costs (MAPNU), where a set of program modules must be assigned to a set of processors. The optimal assignment minimizes the sum of execution costs and communication costs between modules. This problem is naturally formulated as a quadratic 0-1 problem with linear constraints. In this paper, we compare two exact solution methods for this problem. The first method is based on linear programming and Mixed Integer Linear Programming. The second one uses semidefinite programming and Mixed Integer Quadratic Programming. Both of these methods are easy to implement by use of available optimization software. We describe each of these methods and carry out a comparative computational work for instances of MAPNU.

Keywords: Quadratic 0-1 Programming, Linearization Techniques, Convex Quadratic Programming, Semidefinite Programming, Module Allocation Problem Experiments

[1]CEDRIC-CNAM, 292 Rue Saint-Martin, Case I418, F-75141 Paris cedex 03, France.

1 Introduction

Due to the advances in VLSI technology, there has been a proliferation of distributed computing systems in the past few years. The *module llocation problem* in these systems consists of finding a suitable assignment of program modules (or tasks) to processors so that the sum of execution and communication costs is minimized. Several variants of this problem have already been considered, with different assumptions on the architecture of the distributed system or on the structure of costs and feasible solutions. See [1], [2], [3], [7], [13], [8]. In this paper, we consider MAPNU, the Module Allocation Problem with Non-Uniform communication costs. In this problem, no particular assumption is made on the execution and communication costs, ans no capacity constraints are considered.

MAPNU is naturally formulated as a quadratic 0-1 problem with linear constraints. This problem is also known in the literature as *the quadartic semiassignement problem*. In this paper, we compare two exact solution methods for this problem. In the first method, we start by solving a linear program in order to get a reduction [4] of the initial problem. Then, the reduced problem is stated as an equivalent compact mixed integer linear program and solved by a Mixed Integer Linear Programming (MILP) solver.

The second method applies the general algorithm described in [5] for a general quadratic 0-1 problem with linear constraints. We start by solving a semidefinite program in order to reformulate the objective function of the initial problem as a convex quadratic function. The reformulated equivalent problem can then be handled by a Mixed Integer Quadratic Programming (MIQP) solver.

The following of this paper is organized as follows. In Section 2, we state the problem and recall its formulation as a quadratic 0-1 problem (Q01). In Section 3 (resp. 4), we describe the linear (resp. quadratic) programming-based exact solution method. Details of our computational experience are given in Section 5.

2 Problem Statement and Formulation by a Quadratic 0-1 Program

Let $\mathcal{P} = \{p_1, p_2, \ldots, p_P\}$ be a set of P processors, and let $\mathcal{T} = \{t_1, t_2, \ldots, t_T\}$ be a set of T program modules to be assigned to the processors. We denote by q_{tp} $(t = 1, \ldots, T,\ p = 1, \ldots, P)$ the execution cost of module t on processor p, and by $c_{tpt'p'}$ $(t,\ t' = 1, \ldots, T,\ t \neq t', p, p' = 1, \ldots, P)$ the communication cost occuring when modules t and t' are respectively assigned assigned to processors p and p'. Costs q_{tp} and $c_{tpt'p'}$ can be either positive or negative.

A natural mathematical formulation of CMAP can be considered by taking the variables vector $x = (x_{tp})$ $(t = 1, \ldots, T; p = 1, \ldots, P)$ where x_{tp} is equal to 1 if module t is allocated to processor p and is equal to 0 otherwise.

MAPNU can be formulated by the following quadratic 0-1 problem:

$$\text{(Q01):} \quad \min F(x) = \sum_{t=1}^{T}\sum_{p=1}^{P} q_{tp}x_{tp} + \sum_{t=1}^{T-1}\sum_{t'=t+1}^{T}\sum_{p=1}^{P}\sum_{p'=1}^{P} c_{tpt'p'}x_{tp}x_{t'p'} \tag{1}$$

s.t.:

$$\sum_{p=1}^{P} x_{tp} = 1 \qquad t = 1, \ldots, T \tag{2}$$

$$x \in \{0,1\}^{T \times P}$$

Constraints (2) mean that every module t is allocated to exactly one processor. The first part of the objective function (1) is the total execution cost and the second part is the total communication cost.

3 The Linear Programming based method

This method is composed of two phases. The first phase operates a "reduction" of (Q01) in the sense that the objective function is written as a constant plus a set of linear or quadratic terms, all of them having nonnegative coefficients. The reduction algorithm we apply here was presented in [4] and was proved to be optimal in the sense that there exist no other reduction with a higher constant.

The Reduction Phase
Let (D) be the following linear problem:

$$(D) \; : \; \max \sum_{t=1}^{T} \lambda_t$$

s.t.:

$$\begin{array}{ll} \lambda_t - \sum_{t'=1}^{t-1} \beta_{t'tp} - \sum_{t'=t+1}^{T} \beta_{t'tp} + \nu_{tp} = q_{tp} & t = 1, \ldots, T; p = 1, \ldots, P \\ \beta_{tt'p'} + \beta_{t'tp} + \delta_{tpt'p'} = c_{tpt'p'} & 1 \leq t' < t \leq T; p, p' = 1, \ldots, P \\ \nu \geq 0 \;, \quad \delta \geq 0 & \end{array}$$

and let $v(D)$ be its optimal value. The following proposition proves that any optimal solution to (D) provides a reduction of (Q01).

Proposition 1 *[4] For any optimal solution $(\lambda, \beta, \nu, \delta)$ to (D), and for any solution x to (Q01) :*

$$F(x) = v(D) + \sum_{t=1}^{T}\sum_{p=1}^{P} \nu_{tp}x_{tp} + \sum_{t=1}^{T-1}\sum_{p=1}^{P}\sum_{t'=t+1}^{T}\sum_{p'=1}^{P} \delta_{tpt'p'}x_{tp}x_{t'p'} \tag{3}$$

■

Replacing the objective function $F(x)$ by its new expression (3), we obtain a reduced problem $(\mathrm{Q01})_r$ equivalent to (Q01). The next phase will be applied to $(\mathrm{Q01})_r$.

The MILP phase

Here, we use a linearization technique, initially proposed by Glover [11]. This method has then been used for example in [6] for the quadratic knapsack problem. The main idea, applied to problem $(\mathrm{Q01})_r$, is to replace the expression $(\sum_{t'=t+1}^{T} \sum_{p'=1}^{P} \delta_{tpt'p'} x_{tp} x_{t'p'})$ by a unique variable h_{tp}, for $t = 1, \ldots, T-1; p = 1, \ldots, P$.

Let ϕ denote the $T \times P$ vector where $\phi_{tp} = \sum_{t'=t+1}^{T} \underset{p'=1..P}{max} (\delta_{tpt'p'})$ and let (L_ϕ) be the following mixed integer linear problem:

$$(L_\phi) \ : \ \min \Delta(x,h) = v(D) + \sum_{t=1}^{T} \sum_{p=1}^{P} \nu_{tp} x_{tp} + \sum_{t=1}^{T-1} \sum_{p=1}^{P} h_{tp}$$

s.t.:

$$\sum_{p=1}^{P} x_{tp} = 1 \qquad t = 1, \ldots, T$$

$$h_{tp} \geq \sum_{t'=t+1}^{T} \sum_{p'=1}^{P} \delta_{tpt'p'} x_{t'p'} - \phi_{tp}(1 - x_{tp}) \qquad t = 1, \ldots, T-1; p = 1, \ldots, P \quad (4)$$

$$h_{tp} \geq 0 \qquad t = 1, \ldots, T-1; p = 1, \ldots, P \quad (5)$$

$$x_{tp} \in \{0, 1\} \qquad t = 1, \ldots, T; p = 1, \ldots, P$$

for any optimal solution $\tilde{x}$ to $(\mathrm{Q01})_r$, it is possible to build an optimal solution $(\tilde{x}, \tilde{h})$ to (L_ϕ) with $\tilde{h}_{tp} = \sum_{t'=t+1}^{T} \sum_{p'=1}^{P} \delta_{tpt'p'} \tilde{x}_{tp} \tilde{x}_{t'p'}$. Indeed, if $\tilde{x}_{tp} = 1$, take $\tilde{h}_{tp}$ equal to its lowest value satisfying Constraints (4), i.e. $\sum_{p'=1}^{P} \delta_{tpt'p'} \tilde{x}_{t'p'}$. If $\tilde{x}_{tp} = 0$, by definition of ϕ_{tp}, Constraints (4) are dominated by the nonnegativity constraints (6), take $\tilde{h}_{tp} = 0$. Hence, problems (L_ϕ), $(\mathrm{Q01})_r$, and (Q01) have the same optimal values.

Let us summarize the Linear Programming based method to solve (Q01):

1. compute an optimal solution to (D) by a Linear Programming solver
2. compute ϕ as $\phi_{tp} = \sum_{t'=t+1}^{T} \underset{p'=1..P}{max} (\delta_{tpt'p'})$
3. solve (L_ϕ) by a MILP solver.

4 The Quadratic Programming based method

This method also is composed of two phases. The first phase aims at reformulating the objective function of (Q01) by a convex quadratic function. The advantage of this reformulation is that the new equivalent problem has a tractable continuous relaxation, that can be embedded within a branch-and-bound algorithm. Further, available MIQP solvers can handle such problems. We use the reformulation algorithm described in [5] which is valid for any linearly constrained quadratic 0-1 program.

The Reformulation Phase
Let (SD) be the following semidefinite program:

$$(\text{SD}):\quad \min \sum_{t=1}^{T}\sum_{p=1}^{P} q_{tp}x_{tp} + \sum_{t=1}^{T-1}\sum_{t'=t+1}^{T}\sum_{p=1}^{P}\sum_{p'=1}^{P} c_{tpt'p'}X_{tpt'p'}$$

s.t.:

$$\sum_{p'=1}^{P} X_{tpt'p'} = x_{tp} \qquad t=1,\dots,T,\ t'=1,\dots,T,\ p=1,\dots,P \tag{6}$$

$$X_{tptp} = x_{tp} \qquad t=1,\dots,T,\ p=1,\dots,P \tag{7}$$

$$\begin{bmatrix} 1 & x^t \\ x & X \end{bmatrix} \succeq 0 \tag{8}$$

where X is a symmetric $(T \times P)$-matrix. (SD) can be viewed as a particular semidefinite programming relaxation of (Q01). As shown in [5], any optimal solution of its dual problem provides a convex reformulation of the objective function. Let (α^*, u^*) be an optimal solution to the dual problem of (SD). We can build the following quadratic 0-1 problem:

$$(\text{CQ01}):\quad \min F^{\alpha^*,u^*}(x) = F(x) + \sum_{t=1}^{T}\sum_{t'=1}^{T}\sum_{p=1}^{P}\alpha^*_{tpt'}x_{tp}(\sum_{p'=1}^{P}x_{t'p'} - 1) + \sum_{t=1}^{T}\sum_{p=1}^{P}u^*_{tp}(x_{tp}^2 - x_{tp})$$

s.t.:

$$\sum_{p=1}^{P} x_{tp} = 1 \qquad t=1,\dots,T$$

$$x \in \{0,1\}^{T\times P}$$

Problem (CQ01) is obviously equivalent to (Q01). In addition, function F^{α^*,u^*} is convex, and the optimal value of the continuous relaxation of (CQ01) is precisely the optimal value of (SD). Many other reformulations of $F(x)$ by a function of the form $F^{\alpha,u}(x) = F(x) + \sum_{t=1}^{T}\sum_{t'=1}^{T}\sum_{p=1}^{P}\alpha_{tpt'}x_{tp}(\sum_{p'=1}^{P}x_{t'p'} - 1) + \sum_{t=1}^{T}\sum_{p=1}^{P}u_{tp}(x_{tp}^2 - x_{tp})$ can be obtained. The above reformulation F^{α^*,u^*} is optimal in the sense that it provides the higher possible bound by continuous relaxation [5].

The MIQP phase This phase consists in solving (CQ01) by a branch-and-bound algorithm which evaluation procedure is based on continuous relaxation. It can be completely handled by available solvers.

Let us summarize the convex Quadratic Programming based method to solve (Q01):

1. compute an optimal solution (α^*, u^*) to (SD) by a semidefinite programming solver
2. solve (CQ01) by a MIQP solver.

5 Computational results

The Instances
We use the instances provided in [9], where coefficients q_{tp} and $c_{tpt'p'}$ are randomly generated in the interval [-50,50].
Machine and software
Our experiments are carried out on a portable PC with a Pentium IV of 1600 MHz and 1024 MB of RAM.

In order to solve (SD), we use SDP_S, the modeling language for semidefinite programming designed by Delaporte et al. (2002). SDP_S works with SBmethod, an SDP solver that implements the spectral bundle method [12].

We use Cplex 8.1 together with AMPL [10] for solving LPs, MILPs ans MIQPs.

The results

				LP-based method				convex QP-based method			
instance	T	P	opt	bound v(D)	gap%	CPU(s)	nodes	bound v(SD)	gap%	CPU(s)	nodes
tassnu15_5_1	15	5	-1985	-2946	48%	13	30892	-2361,7	19%	8	2865
tassnu15_5_2	15	5	-1568	-2754,1	76%	19	55301	-2154,6	37%	23	27953
tassnu15_5_3	15	5	-1892	-2946,2	56%	24	55914	-2430,6	28%	16	12454
tassnu15_5_4	15	5	-1806	-2729,2	51%	14	34661	-2349	30%	12	9509
tassnu15_5_5	15	5	-1881	-2778,3	48%	12	28803	-2360	25%	26	6853
averages					**56%**	**17**	**41114**		**28%**	**17**	**11927**
tassnu18_4_1	18	4	-2136	-3650	71%	50	106172	-2656,6	24%	25	11268
tassnu18_4_2	18	4	-1936	-3458,5	79%	38	76806	-2358,3	22%	33	3082
tassnu18_4_3	18	4	-2304	-3759,25	63%	13	28796	-2741,3	19%	16	1906
tassnu18_4_4	18	4	-1926	-3535	84%	40	88725	-2519,3	31%	13	13578
tassnu18_4_5	18	4	-2044	-3506,25	72%	24	64407	-2413,2	18%	14	2893
averages					**74%**	**33**	**72981**		**23%**	**20**	**6545**
tassnu20_5_1	20	5	-2587	-5039,6	95%	1672	3151497	-3561	38%	320	317602
tassnu20_5_2	20	5	-2810	-5154,6	99%	-	3303233	-3781,2	46%	392	420129
tassnu20_5_3	20	5	-2512	-4981	77%	-	2582615	-3528,8	26%	756	859562
tassnu20_5_4	20	5	-2777	-5198,2	107%	1487	2727249	-3656,3	46%	413	463540
tassnu20_5_5	20	5	-2744	-5067,2	82%	1765	2651618	-3687,8	33%	223	238692
averages					**92%**	**1641**	**2883242**		**38%**	**421**	**459905**

Table 1: Exact solution of instances from [9]
- : the branch-and-bound algorithm did not stop after 30' of CPU time

Table 5 gives the results of our two exact methods on a selection of instances from [9]. In the first columns columns are the name of the instance, the number of tasks T, the number of processors P and the optimal values of the instance. The four following columns contain the results of the LP-based method and the last four columns contain the results of the QP-based method. Columns gap give the gaps associated to bound $v(D)$ in the LP-based method and the bound $v(SD)$ in the QP-based method. We can observe that this gap is much better in the QP-based method. This explains in part the better general behavior of this last method.

References

[1] R. K. Arora and S. P. Rana. Analysis of the module assignment problem in distributed computing systems with limited storage. *Information Processing Letters*, 10(3):111–115, April 1980.

[2] A. Billionnet, M. C. Costa, and A. Sutter. An efficient algorithm for a task allocation problem. *Journal of the ACM*, 39(3):502–518, July 1992.

[3] A. Billionnet and S. Elloumi. Placement de tâches dans un système distribué et dualité lagrangienne. *Revue d'Automatique, d'Informatique et de Recherche Opérationnelle (R.A.I.R.O.), série verte*, 26(1):83–97, 1992.

[4] A. Billionnet and S. Elloumi. Best reduction of the quadratic semi-assignment problem. *DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 109:197–213, 2001.

[5] A. Billionnet, S. Elloumi, and M.C. Plateau. Convex quadratic programming for exact solution of 0-1 quadratic programs. Technical Report 000, CEDRIC, 2005.

[6] A. Billionnet and E. Soutif. Using a mixed integer programming tool for solving the 0-1 quadratic knapsack problem. Forthcoming in INFORMS Journal on Computing.

[7] Maw-Sheng S. Chern, G. H. Chen, and Pangfeng Liu. An LC branch-and-bound algorithm for the module assignment problem. *Information Processing Letters*, 32(2):61–71, July 1989.

[8] W. Fernandez de la Vega and M.Lamari. The task allocation problem with constant communication. *Discrete Applied Mathematics*, 2003. In Press.

[9] S. Elloumi. The task assignment problem, a library of instances. *http://cedric.cnam.fr/oc/TAP/TAP.html*.

[10] R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. The Scientific Press (now an imprint of Boyd & Fraser Publishing Co.), Danvers, MA, USA, 1993.

[11] F. Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22:455–460, 1975.

[12] C. Helmberg. *A C++ implementation of the Spectral Bundle Method*. http://www.mathematik.uni-kl.de/helmberg/SBmethod. Version 1.1.

[13] F. Roupin. On approximating the memory-Constrained Module Allocation Problem. *Information Processing Letters*, 61(4):205–208, March 1997.