Two pricing methods for solving an integrated commercial fishery planning model

MB Hasan[*]   JF Raffensperger[†]

Abstract

In this paper, we develop two novel pricing methods for solving an integer program. We demonstrate the methods by solving an integrated commercial fishery planning model (IFPM). In this problem, a fishery manager must schedule fishing trawlers (determine when and where the trawlers should go fishing, and when the trawlers should return the caught fish to the factory). The manager must then decide how to process the fish into products at the factory. The objective is to maximise profit. The problem may be modelled as a single integer program, with both the trawler scheduling and production planning parts integrated. Inventory constraints connect the two parts of the problem. Production planning alone would result in an easy linear program, but due to the trawler scheduling aspect, the IFPM is a hard integer program in the sense that traditional solution methods result in computation times that are far too long to be practical. The two pricing methods developed in this paper are a decomposition–based O'Neill pricing method and a reduced cost–based pricing method. We demonstrate the methods by means of numerical examples for different planning horizons, corresponding to differently sized problems.

Key words: Decomposition, pricing, reduced cost, fishing trawler scheduling.

1 Introduction

In this paper we present recent research on the solution of an integer program for an integrated commercial fishery's activities. Two loosely-connected problems arise in a modern commercial fishery. The first is to schedule trawlers for fishing, including deciding where and when those trawlers should work, and, crucially, when they should return to land the fish. The landed fish generally becomes inventory, which is raw material for a processing plant. The processing plant cleans, processes, and packages the fish for the market. The second problem is to schedule the processing of different types of products.

Based on real data for a commercial fishery in New Zealand, we previously developed a model (Hasan and Raffensperger, 2006) to solve this problem: the *integrated fishery*

---

[*]Corresponding author: Department of Management, University of Canterbury, Private Bag 4800, Christchurch, 8020, New Zealand, email: b.hasan@mang.canterbury.ac.nz

[†]Department of Management, University of Canterbury, Private Bag 4800, Christchurch, 8020, New Zealand.

planning model (IFPM). The IFPM is designed to co-ordinate trawler scheduling and processing. The model can theoretically be updated and solved periodically to aid in a manager's decision making. Unfortunately, for realistic planning horizons of 20 periods or more, computational times involved in solving the IFPM are quite long, even with methods such as Dantzig-Wolfe decomposition and subgradient optimisation. We have since developed two novel column generation algorithms to solve the IFPM. These algorithms show promise and are based on the decomposition-based pricing algorithm of Mamer and McBride (2000), combined with the integer variable pricing method of O'Neill *et al.* (2005).

1.1   The fishery planning literature

Wide-ranging research has been reported on fisheries. Many papers describe biological models, but only a few consider production planning. Mikalsen and Vassdal (1981) developed a multi-period linear programming (LP) model for one month production planning so as to smooth the seasonal fluctuations of fish supply. Their model is market-driven and incorporates the acquisition of raw material purchased (rather than acquired with their own fishing fleet).

Jensson (1988) developed a product mix LP model to maximize profit of an Icelandic fish processing firm over a five period planning horizon. He addressed both production planning and labour allocation for that processing firm, but did not consider any fleet-specific issues or quota sizes.

Gunn *et al.* (1991) developed a model for calculating the total profit of a Canadian company with respect to integrated fishing and processing. Their model includes a fleet of trawlers, a number of processing plants and market requirements. However, their model ignores the trawler scheduling and labour allocation in the processing firm. Indeed, none of these papers report models that attempt to integrate both trawler scheduling and production.

1.2   The integer programming literature

The literature on integer programming is extensive. We describe only a few papers here that have informed our work.

Martin *et al.* (1985) presented a reduced cost-based branch-and-bound method for solving mixed integer linear programs (MILPs). The authors formulate two candidate problems on the basis of 0-1 integer variables and then optimize both of the candidate problems in order to obtain the MILP solution.

Mamer and McBride (2000) developed a *decomposition-based pricing* (DBP) procedure for linear programs (LPs). Their algorithm works by solving subproblems just as the Dantzig-Wolfe algorithm uses subproblems. However, the DBP master problem exhibits the same form and structure as the original model, but with far fewer variables. Variables that are positive in the subproblem are brought directly into the master problem; all other variables are omitted from the master problem. Our work in this paper builds substantially on the ideas of Mamer and McBride (2000). A DBP approach has also

been adopted by De Carvalho (2006) for cutting stock, and by Raffensperger and Schrage (2007) for scheduling training in a tank battalion. We have also previously adopted a DBP approach with respect to an IFPM in Hasan and Raffensperger (2007). In this paper, we describe two improved DBP methods.

The first method we call *decomposition-based O'Neill pricing* (DBONP), because it is based on the work of O'Neill *et al.* (2005). These authors developed a technique for constructing a set of linear prices by solving a MILP and an associated LP, based on a theorem of Gomory and Baumol (1960). They first solve a MILP, set the integer variables to their optimal values, and then remove the integrality constraints to convert the MILP to an LP. They use the dual prices obtained from this LP to form an efficient contract (the dual of the *IFPM*) in the context of an electricity market.

The second method is a *reduced cost-based pricing* (RCBP) method. Unlike Martin *et al.* (1985), we set constraints for both 0-1 integer variables (O'Neill *et al.*, 2005) in the same candidate problem, which is the restricted master problem in the proposed RCBP method. In this method, we do not solve a subproblem at all. Instead, we choose new variables for the restricted master problem based on a reduced cost calculation, and we bring a set of variables into the restricted master problem at each iteration. We show that both of these methods produce better solutions than those reported in our earlier work (Hasan and Raffensperger, 2007).

The remainder of this paper is organized as follows. In §2, we briefly present the IFPM. In §3, we review O'Neill's pricing method and describe the mathematical formulation of the proposed DBONP method. We also present the DBONP algorithm along with numerical examples. In §4, we present the mathematical formulation of the proposed RCBP method, and also present the RCBP algorithm along with numerical examples. The solutions obtained by the DBONP and RCBP methods are compared with that of DBP in §5. Some conclusions follow in §6.

2   The fishery model in matrix notation

In this section we briefly describe our IFPM. The details of the model can be found in Hasan and Raffensperger (2006). We have omitted details of the model in order to focus on the algorithm.

Parameters
Let $V_{t,v}$ be the profit earned by trawler v during period t and let I_t denote the cost per kg of fish landed during period t. Furthermore, suppose $P_{i,j,l}$ is the price per kg of fish type i converted into product j of quality l and let $A^{(0)}_{a,i,t,v}$ denote the amount of raw fish i that will be landed during period t by trawler v from area a. Also, let $A^{(1)}_{i,l,t,v}$ denote the mass transformation for raw fish type i from trawler v of quality l during period t, and let $A^{(2)}_{i,j,l,t}$ denote the mass transformation for raw fish type i into finished product type j of quality l during period t. Finally, let $d^{(1)}_{a,t,v}$ denote the mass balance coefficients on trawler v during period t in area a and let $d^{(2)}_{i,j,l,t}$ denote the mass transformation for raw fish type i into finished product type j of quality l during period t.

Decision variables

Let $w_{p,a,u,t,v}$ be a binary decision variable taking on the value of 1 if a trawler v will go fishing during period u in area a from factory p and lands its catch during period t, or 0 otherwise. Furthermore, let the variable $f_{i,l,t}$ denote the current quantity of fish type i of quality l during period t and let the variable $x_{i,j,l,t}$ denote the amount of fish type i converted into product j of quality l during period t.

The objective in the IFPM is to

$$\text{maximize} \quad \sum_{t,v} V_{t,v} \sum_{p,a,u} (u-t) w_{p,a,u,t,v} - \sum_{t} I_t \sum_{i,l} f_{i,l,t} + \sum_{i,j,l} P_{i,j,l} \sum_{t} x_{i,j,l,t},$$

subject to

$$\sum_{p,u,v} A^{(0)}_{a,i,t,v} w_{p,a,u,t,v} = f_{i,l,t} \qquad \text{for all } i,l,t, \tag{1}$$

$$\sum_{a,p,u,v} d^{(1)}_{a,t,v} w_{p,a,u,t,v} = b^{(1)}_t \qquad \text{for all } t, \tag{2}$$

$$\sum_{j} d^{(2)}_{i,j,l,t} x_{i,j,l,t} = b^{(2)}_{i,l,t} \qquad \text{for all } i,l,t, \tag{3}$$

$$\sum_{v} A^{(1)}_{i,l,t,v} f_{i,l,t} + \sum_{j} A^{(2)}_{i,j,l,t} x_{i,j,l,t} = b^{(0)}_{i,l,t} \qquad \text{for all } i,l,t, \tag{4}$$

$$w_{p,a,u,t,v} \in \{0,1\}, \qquad \text{for all } p,a,u,t,v \tag{5}$$

$$f_{i,l,t}, x_{i,j,l,t} \geq 0 \qquad \text{for all } i,j,l,t. \tag{6}$$

where $b^{(0)}_{i,l,t}$ denotes the restriction on the quantity of fish type i of quality l during period t, $b^{(1)}_t$ denotes the restriction on the trawler scheduling constraint and $b^{(2)}_{i,j,t}$ denotes the restriction on the quantity of fish type i converted into product j during period t.

Constraint set (1) represents the relationship of the trawler scheduling variables w to landed fish f, as a mass balance in movement of fish from trawlers to the factory, while (2) expresses constraints involving only trawler scheduling, indicating, for example, that a trawler may be in only one place at a time. Constraint set (3) expresses fish processing restrictions, modelling the flow of fish through the factory as raw fish is converted into various products. Constraint set (4) constitutes mass balance constraints, representing the flow of raw landed fish inventory into the fish processing factory. When the integer constraints (5) are relaxed, the model is the usual linear programming relaxation.

The IFPM consists of trawler scheduling and processing, connected by inventory constraints, either (1) or (4). Using Lagrangean relaxation, one can relax either of these side constraints, in which case the model decomposes into an integer program for trawler scheduling, and a linear program for the fish processing. These separate problems are easier to solve, and the sum of their objective values represents an upper bound (since it is a maximization problem) on the optimal objective function value of the IFPM. For

example, if we relax (4), we obtain the two subproblems,

$$
\left.\begin{aligned}
\text{maximize} \quad & \sum_{i,j,l} P_{i,j,l} \sum_{t} x_{i,j,l,t} - \sum_{i,l,t} \theta_{i,l,t} \sum_{j} A^{(2)}_{i,j,l,t} x_{i,j,l,t} \\
\text{subject to} \quad & \sum_{j} d^{(2)}_{i,j,l,t} x_{i,j,l,t} = b^{(2)}_{i,l,t} \qquad \text{for all } i, l, t, \\
& x_{i,j,l,t} \geq 0, \qquad\qquad\qquad \text{for all } i, j, l, t,
\end{aligned}\right\} P_1(\boldsymbol{\theta}) \qquad (7)
$$

and

$$
\left.\begin{aligned}
\text{maximize} \quad & \sum_{t,v} V_{t,v} \sum_{p,a,u} (u - t) w_{p,a,u,t,v} - \sum_{t} I_t \sum_{i,l} f_{i,l,t} \\
& \qquad\qquad - \sum_{i,l,t} \theta_{i,l,t} f_{i,l,t} \sum_{v} A^{(1)}_{i,l,t,v} \\
\text{subject to} \quad & \sum_{p,u} A^{(0)}_{a,i,t,v} w_{p,a,u,t,v} = f_{i,l,t} \qquad \text{for all } i, l, t, \\
& \sum_{a,p,u,v} d^{(1)}_{a,t,v} w_{p,a,u,t,v} = b^{(1)}_t \qquad \text{for all } t, \\
& w_{p,a,u,t,v} \in \{0, 1\} \qquad\qquad \text{for all } p, a, u, t, v
\end{aligned}\right\} P_2(\boldsymbol{\theta}), \qquad (8)
$$

where $\boldsymbol{\theta} = [\theta_{i,l,t}]$ is the matrix of dual prices on (4) assuming that the IFPM is solved as an LP, not as an integer program (IP). It is unfortunate that $P_2(\boldsymbol{\theta})$ cannot be directed by some kind of standard price information on the integer variable $\boldsymbol{w} = [w_{p,a,u,t,v}]$. In fact, the DBONP method actually finds such price information, and uses it.

Following the decomposition-based pricing method for this problem (Hasan and Raffensperger, 2007), the master problem follows from the original problem, assuming its structure and including all its constraints. However, initially only enough variables are included to allow a feasible solution. In the IFPM, the zero matrix is feasible, as the fishery manager can simply choose to do nothing.

At iteration k, the master problem M^k is solved as a linear program, in order to find the necessary dual prices $\boldsymbol{\theta}$. These prices are passed to the subproblems $P_1(\boldsymbol{\theta})$ and $P_2(\boldsymbol{\theta})$, which are then solved. Positive variables from the subproblems are then passed to the master problem, increasing the total number of variables that it contains. The set of variables in the master problem is tracked by the indices of the positive variables found thus far, in an index set I^k. Thus, a variable with its index in I^k has been positive in a subproblem in some previous iteration, and will appear in the master problem. Variables that have always been zero in every subproblem do not have their index in I^k, and thus do not appear in the master problem. In the master problem M^k the objective is to

$$
\text{maximize} \quad \sum_{t,v} V_{t,v} \sum_{p,a,u} (u - t) w_{p,a,u,t,v} - \sum_{t} I_t \sum_{i,l} f_{i,l,t} + \sum_{i,j,l} P_{i,j,l} \sum_{t} x_{i,j,l,t},
$$

$$\text{subject to} \quad \sum_{p,u,v} A^{(0)}_{a,i,t,v} w_{p,a,u,t,v} = f_{i,l,t} \quad \text{for all } i,l,t,$$

$$\sum_{a,p,u,v} d^{(1)}_{a,t,v} w_{p,a,u,t,v} = b^{(1)}_t \quad \text{for all } t,$$

$$\sum_j d^{(2)}_{i,j,l,t} x_{i,j,l,t} = b^{(2)}_{i,l,t} \quad \text{for all } i,l,t,$$

$$\sum_v A^{(1)}_{i,l,t,v} f_{i,l,t} + \sum_j A^{(2)}_{i,j,l,t} x_{i,j,l,t} = b^{(0)}_{i,l,t} \quad \text{for all } i,l,t,$$

$$w_{p,a,u,t,v}, f_{i,l,t}, x_{i,j,l,t} \geq 0, \quad \text{for all } p,a,u,t,v,i,l,t,j$$

$$f_{i,l,t}, w_{p,a,u,t,v}, x_{i,j,l,t} \in I^k \quad \text{for all } p,a,u,t,v,i,l,t,j,$$

where I^k is the index set of positive variables found in the subproblems, and where $f_{i,l,t}$, $w_{p,a,u,t,v}$, $x_{i,j,l,t} = 0$ for $f_{i,l,t}, w_{p,a,u,t,v}, x_{i,j,l,t} \notin I^k$. The index set I^k increases in size with each iteration because each iteration of the subproblems adds new positive variables.

While this decomposition approach is already better than a direct integer programming approach using CPlex, for example, we wished to improve the method further.

3 Decomposition-based O'Neill pricing (DBONP)

In this section, we first discuss the notion of O'Neill pricing in §3.1. In §3.2, we then present the mathematical formulation of the DBONP and present the DBONP algorithm. This is followed by a presentation of numerical examples over different planning horizons §3.3.

3.1 O'Neill's pricing method

O'Neill *et al.* (2005) developed a technique for constructing a set of linear prices from solving an MILP and an associated LP, based on the following theorem of Gomory and Baumol (1960).

Theorem 1 *An MILP with m continuous variables and n integer variables that has a feasible and bounded optimal solution in $(\mathbb{R}^m \times \mathbb{Z}^n)$ can be converted to an LP with at most $(m + n)$ continuous variables and at most n additional linear constraints.* ∎

These authors were not interested in a solution as such, nor in the associated computation time, but in finding efficient prices for indivisible objects. To find these prices, they first solved an MILP to optimality. They then added new constraints that fix the integer variables to their optimal values, and removed the integrality constraints to convert the MILP to an LP. Solution of this problem gave dual prices to the new constraints. They showed that the dual variables in the LP have a traditional interpretation as prices. The dual variables explicitly price integral activities, and clear the market in the presence of nonconvexities. They used these dual prices to form an efficient contract in the context of a market for electricity.

3.2   Mathematical formulation for DBONP

To apply the method of O'Neill *et al.* (2005) in the context of DBP, our method

1. finds an optimal solution $\boldsymbol{w} = [w_{p,a,u,t,v}]$ to the restricted master problem as an integer program;
2. fixes the integer variables to their optimal values $\boldsymbol{w}^*$ by means of new constraints of the form $\boldsymbol{w} = \boldsymbol{w}^*$ and solves the restricted master problem as an LP, thus obtaining dual price information $\boldsymbol{\theta_1}$ on the constraints in (9);
3. then uses the resulting dual prices $\boldsymbol{\theta_1}$ to better inform the trawler scheduling sub-problem as to which variables should be selected. The trawler subproblem can use this new information through Lagrangean relaxation of the new constraints, that is by solving the following problem, called $P(\boldsymbol{\theta}, \boldsymbol{\theta}_1)$, in which the objective is to

$$
\left.
\begin{aligned}
\text{maximize} \quad & \sum_{t,v} V_{t,v} \sum_{p,a,u} (u-t) w_{p,a,u,t,v} - \sum_t \sum_{i,l} f_{i,l,t} \\
& - \sum_{i,l,t} \theta_{i,l,t} \sum_v A^{(1)}_{i,l,t,v} \\
& - \sum_{i,l,t} {}^1\theta_{i,l,t}(u-t) \sum_{p,a,u} (w_{p,a,u,t,v} - w^*_{p,a,u,t,v}) \\
\text{subject to} \quad & \sum_{p,u} A^{(0)}_{a,i,t,v} w_{p,a,u,t,v} = f_{i,l,t} \quad \text{for all } i,l,t, \\
& \sum_{a,p,u,v} d^{(1)}_{a,t,v} w_{p,a,u,t,v} \leq {}^1 b_t \quad \text{for all } t, \\
& w_{p,a,u,t,v} \in \{0,1\}, \quad \text{for all } p,a,u,t,v \\
& f_{i,l,t} \geq 0, \qquad \text{for all } i,l,t.
\end{aligned}
\right\} P(\boldsymbol{\theta}, \boldsymbol{\theta_1})
$$

4. Positive variables from both subproblems are brought into the restricted master problem. Two stopping criteria are enforced, namely when no new positive variables are produced, or when the objective values of the subproblems and master problem are equal. By explicitly pricing the integer variables, and using that price information in the subproblem, we bring better variables into the restricted master problem, and return to step 1.

Note, however, that this approach requires solving the restricted master problem as an integer program at every iteration. This is computationally expensive. We therefore employ ordinary DBP, solving the restricted master problem and subproblems as LPs, until we find an LP optimum. We then switch to the formal DBONP method, and continue iterating. This approach creates two separate loops. The first loop does not utilize the constraints (9), whereas the second loop does.

Loop 1. Relax the inventory balance constraint (4), and then apply the DBP method, to obtain the final restricted master problem as an LP.

    Step 0: Initialize. Set iteration number $k \leftarrow 1$ and the initial prices $\boldsymbol{\theta}_1 \leftarrow \boldsymbol{\theta}$.

Step 1: Solve subproblems $P_1(\boldsymbol{\theta})$ and $P_2(\boldsymbol{\theta})$, treating $P_2(\boldsymbol{\theta})$ as an IP. For $\boldsymbol{w}^i > \mathbf{0}$ put i in I^k, where $I^k = \{i : \boldsymbol{w}^i > \mathbf{0}$ in $P_1(\boldsymbol{\theta})$, and $P_2(\boldsymbol{\theta})$ for any iteration $1, 2, \ldots, k\}$.

Step 2: Solve M^k as an LP to obtain dual prices $\boldsymbol{\theta}^k$ and pass them to the subproblems.

Step 3: If $v\left(P_1(\boldsymbol{\theta}) + P_2(\boldsymbol{\theta})\right) = v\left(M^{k+1}\right)$, then go to Loop 2. Else $k \leftarrow k + 1$ and go to step 1. Here $v\left(P_1(\boldsymbol{\theta}) + P_2(\boldsymbol{\theta})\right)$ represents the objective function value of the subproblems and $v\left(M^{k+1}\right)$ represents the objective function value of the restricted master problem.

Loop 2. Solve the current restricted master problem as an IP, and add constraints which fix the integer variables to their optimal values. Solve the master problem as an LP and obtain the dual prices on the inventory balance constraint (4), and the equations associated with the integer variables. We have the dual prices $\boldsymbol{\theta}^k$ as before, but now we also have new dual prices $\boldsymbol{\theta}_1$ from the new constraints.

Step 4: Solve the restricted master problem as an IP.

Step 5: For integer variables, fix $\boldsymbol{w}^i = \boldsymbol{w}^{i*}$.

Step 6 : Solve the master problem as an LP with $\boldsymbol{w}^i$ fixed. Obtain dual prices $\boldsymbol{\theta}^k$ and $\boldsymbol{\theta}_1$, and pass them to the subproblems.

Step 7: Solve the subproblems $P_1(\boldsymbol{\theta})$ and $P_2(\boldsymbol{\theta}, \boldsymbol{\theta}_1)$ with the dual prices obtained from step 6. If no new variables enter into the restricted master problem, then stop. Else go back to step 4.

We present the logic of the DBONP algorithm in the form of a flowchart Figure 1.

3.3 Numerical results

We compare the solutions of the DBONP approach with those obtained from the original IFPM, LP relaxation problem, and DBP algorithm. The results are presented in Table 1. We observe no duality gap for the 5, 10 and 25-period models, thus confirming optimality. However, the 15, 20, and 30-period models exhibit small gaps. For example, a 30-period model exhibits a 0.02% duality gap. The average duality gap is only 0.04% computed over the six different planning horizon models. These gaps may be considered negligible. Notice that for the results described above we started with dual prices of $\boldsymbol{\theta} = \mathbf{0}$. Instead, we also attempted creating the initial dual prices naively. Results are reported in Table 2. Solutions obtained from DBONP are close to the true optima. The average duality gap is only 0.06%, but sometimes worse than in Table 1.

Tables 1 and 2 show that the solutions obtained via the DBONP approach are either equal to or very close to the optimal solutions (15-period, 20-period and 30-period models). To see why a small difference in profit remains, we compared the true optimal solution with that obtained by the DBONP algorithm for a 30-period planning horizon problem. The total number of trawler trips in the DBONP solution coincides with that in the exact solution, but the schedule is slightly different, as shown in Figures 2 and 3. As a result, there is a slight change in the processing and holding costs.
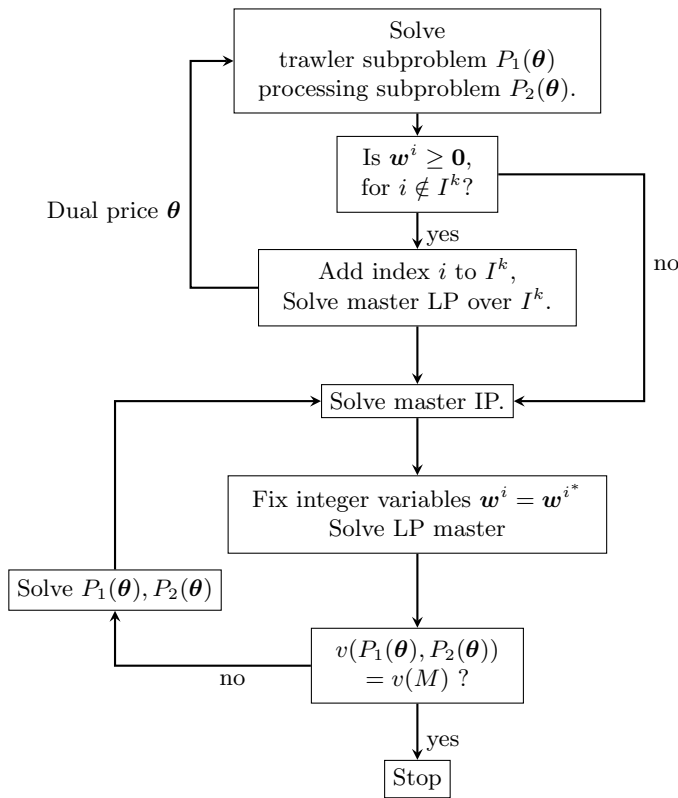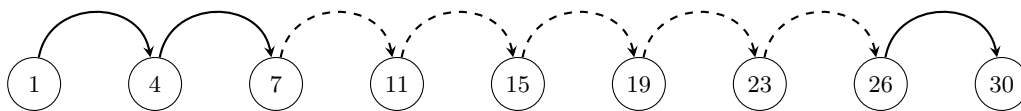
Figure 1: *Flowchart of the DBONP procedure.*
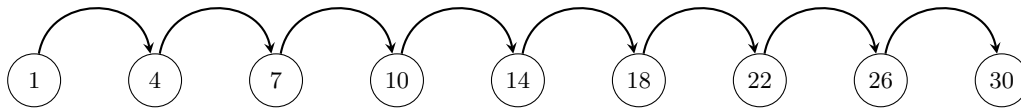
Figure 2: *Trawler 1 schedule in the optimal solution. Here the edges represent periods and vertices represent the required number of periods for a trip.*

Figure 3: *Trawler 1 schedule in the DBONP solution. Here the normal edges represent the trawler trips which coincide with the schedule obtained by the exact (IP) solution and the dashed edges represent the trips which are slightly different from the schedule obtained by the exact (IP) solution.*

Figures 4, 5, and 6 show the solution times, duality gap, and number of iterations, for different planning horizon models respectively, when solved by the DBP and DBONP algorithms. The DBONP approach requires a larger number of iterations and more computation time, but produces better solutions than the DBP approach.

In this section we employed both DBP and O'Neill pricing to develop the DBONP tech-

Length of planning Horizon	Number of variables	Number of iterations	Solution time (s)	DBP solution	DBONP solution	Duality gap
5	489	29	217	$522 764	$522 764	0.00%
10	1 284	27	216	$1 065 540	$1 065 775	0.00%
15	2 229	33	345	$1 579 309	$1 579 570	0.15%
20	3 324	48	912	$1 874 097	$1 878 580	0.08%
25	6 440	45	796	$2 120 282	$2 121 887	0.00%
30	6 938	44	3 562	$2 293 803	$2 300 230	0.02%

Table 1: *Comparison of the solutions obtained by the DBP and DBONP methods. All computations performed on a Pentium III processor with a clock speed of 665 MHz and 384 MB RAM.*

Length of planning Horizon	Number of variables	Number of iterations	Solution time(s)	DBP solution	DBONP solution	Duality gap
5	1 264	29	208	$522 764	$522 764	0.00%
10	2 601	30	266	$1 065 540	$1 065 540	0.02%
15	4 087	36	387	$1 579 309	$1 580 670	0.08%
20	4 926	50	1045	$1 874 097	$1 873 950	0.30%
25	6 259	43	710	$2 120 282	$2 121 887	0.00%
30	8 277	50	3129	$2 293 803	$2 300 460	0.01%

Table 2: *Comparison of the number of iterations, computation times and solutions obtained by the DBP and DBONP methods. All computations performed on a Pentium III processor with a clock speed of 665 MHz and 384 MB RAM.*

nique. We found that the DBONP algorithm requires slightly longer computation times, but produces better solutions than our earlier DBP procedure. To improve further on the computation times, we also developed a reduced cost–based pricing method.

## 4 The Reduced cost–based pricing for IFPM

One reason why the DBONP algorithm took a relatively long computation time was due to solution of the trawler scheduling subproblem as an IP. We therefore attempted to eliminate this step. Instead, we use the O'Neill price information to find the reduced cost for each integer variable. Under this approach we are moving away from the DBP philosophy for the trawler scheduling aspect of the problem, but we continue to use DBP for the fish processing subproblem. So the processing subproblem, and the restricted master problem, are the same as with the DBP approach. Instead of employing the trawler scheduling subproblem, we merely calculate the reduced cost of the variables of that subproblem, which is extremely fast.

4.1 The Reduced cost of a variable

The reduced cost of a variable w_j with associated objective function coefficient c_j is the net change in the objective function when generating one unit of w_j, and is defined as
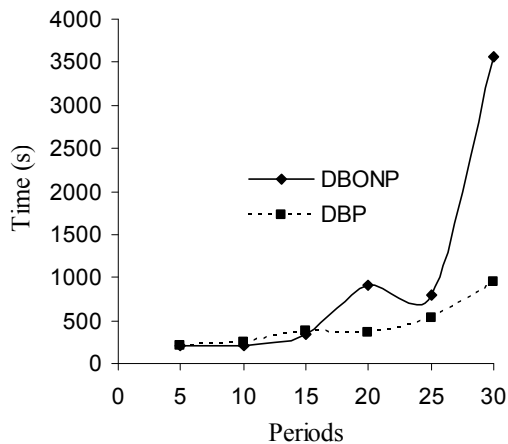
Figure 4: *Solution times required by DBP and DBONP for different planning horizons. All computations performed on a Pentium III processor with a clock speed of 665 MHz and 384 MB RAM.*
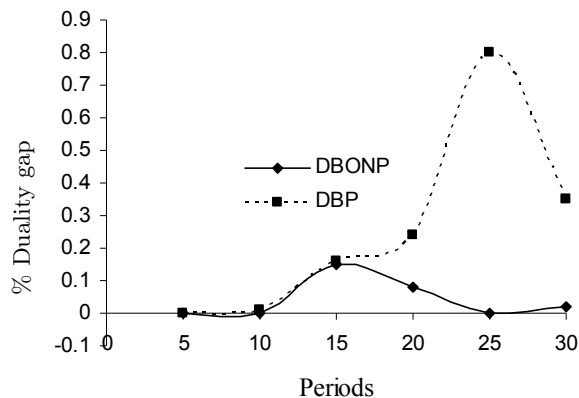

Figure 5: *% Duality gap of DBP and DBONP.*

$\bar{c}_j = c_j - z_j$, where z_j denotes $\boldsymbol{c}_{BV} B^{-1} \boldsymbol{a}_j$. Here $\boldsymbol{c}_{BV}$ are the cost coefficients of the basic variables, B^{-1} is the inverse of the basis matrix, and a_j is the corresponding column of the basic variables. The reduced cost gives the marginal value of a variable on the objective function related to the current basic solution. For a maximization problem, the variable with largest positive reduced cost will be the incoming variable. Following the notation in AMPL (Fourer *et al.*, 1993), we denote the reduced cost of variable $\boldsymbol{w}$ as $\boldsymbol{w}.rc$. Denote $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ as the dual prices on (1) and (2) respectively, with $\boldsymbol{a}_0$ and $\boldsymbol{d}_1$ as the relevant columns of A_0 and D_1 respectively. Then

$$\boldsymbol{w}.rc = \boldsymbol{c}_1 - \boldsymbol{\lambda}_1 \boldsymbol{a}_0 - \boldsymbol{\lambda}_2 \boldsymbol{d}_1 - \boldsymbol{\theta}' \qquad (9)$$

This reduced cost calculation has an explicit term for the integrality constraint. In the next section, we show how to use this reduced cost calculation.
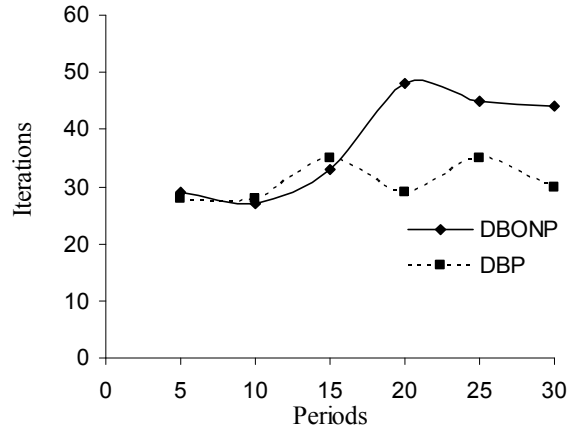
Figure 6: *Number of iterations required to solve different planning horizons by DBP and DBONP.*

4.2 The RCBP algorithm

The RCBP algorithm proceeds as follows:

Step 0. Set $k \leftarrow 1$.

Step 1. Solve M^k as an IP.

Step 2. Add constraints of the form (9) for the integer variables. Solve the restricted master problem as an LP. Obtain dual prices for the trawler scheduling constraints (1), (2), and (9).

Step 3. Calculate $\boldsymbol{w}.rc$ in (11). Scan the reduced costs for all integer variables. Include integer variables with positive reduced cost to the restricted master problem. For the continuous variables from the fish processing part of the problem, there are two options:

> **Option 1:** All continuous variables appear in every restricted master problem.
>
> **Option 2:** Continuous variables with positive reduced cost are added to the restricted master problem at each iteration.

Step 4. For the processing subproblem, solve the processing LP subproblem, and add all positive variables to the restricted master problem as in the DBP approach.

Step 5. If no new variable enters the restricted master problem, then stop. Else $k \leftarrow k+1$ and go back to step 1.

We present the logic of the RCBP algorithm in the form of a flowchart in Figure 7.

4.3 Numerical results

We solved IFPM with different planning horizon models using each option in Step 3. Option 2 takes fewer iterations and less time to solve the fishery model than does Option 1. Results are reported in Table 3.
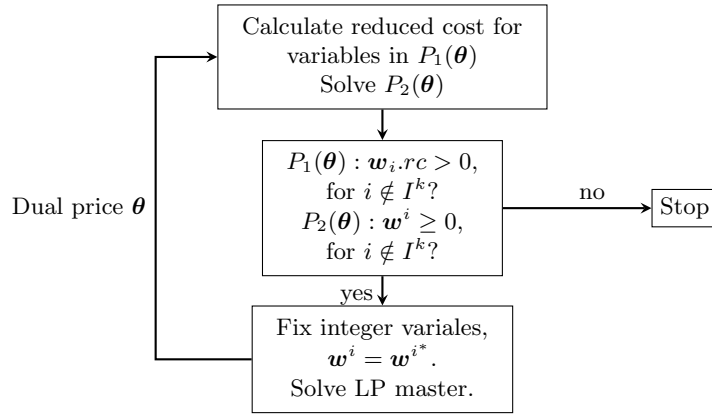
Figure 7: *Flowchart of the RCBP procedure.*

| Planning Horizon	Description of entering variables	Number of iterations	Solution time (sec.)	RCBP optimal value	Duality gap
5	Option 1	5	39	$522 764	0%
	Option 2	3	5	$522 764	0%
10	Option 1	10	142	$1 065 538	0.02%
	Option 2	5	15	$1 065 538	0.02%
15	Option 1	11	113	$1 582 006	0%
	Option 2	5	53	$1 582 008	0%
20	Option 1	7	109	$1 877 275	0.15%
	Option 2	4	71	$1 879 928	0.01%
25	Option 1	6	74	$2 107 736	0.66%
	Option 2	8	111	$2 121 887	0%
30	Option 1	8	262	$2 284 545	0.71%
	Option 2	10	901	$2 299 648	0.05%

Table 3: *Total profit, number of iterations, and solution times for the RCBP procedure.*

5 Comparison of DBP, DBONP and RCBP solutions

In this section we compare the solutions obtained as well as the number of iterations and solution times required by the DBP, DBONP and RCBP methods in Figures 8–10. The RCBP algorithm is the best among the methods we developed. It requires the smallest solution time to solve, requires fewer iterations and yields better solutions. We further investigated these methods using three different problem instances under many different catch rate scenarios. The numerical results reported here are consistent with those otained for the other problem instances.
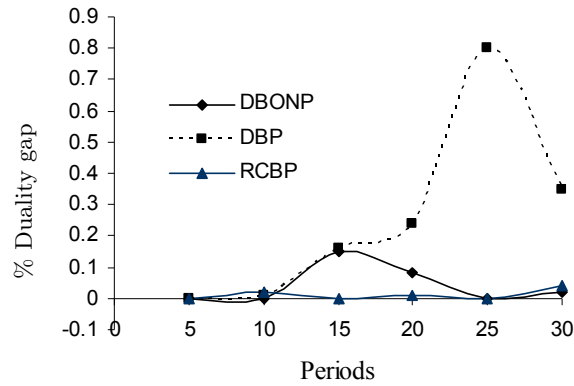
Figure 8: *% Duality gap of DBP, DBONP, and RCBP.*
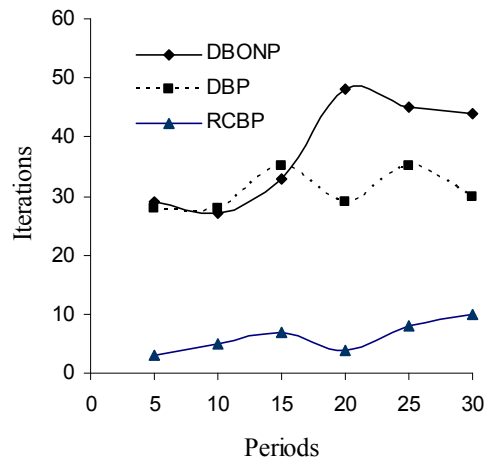
**Figure 9:** *Number of iterations required to solve DBP, DBONP, and RCBP.*

6 Conclusion

In this paper we developed two different column generation algorithms for faster solution of an IFPM. The first is the DBONP method and the second is the RCBP method — both these approaches are based on O'Neill pricing.

In the RCBP method, we solved only easy LP subproblems and avoided the need to solve IP subproblems. Instead of solving the IP trawler scheduling subproblem, we calculated the reduced cost for each variable, choosing variables with positive reduced cost to bring into the restricted master problem.

Compared to the DBP method alone, we found that the DBONP algorithm took slightly longer, but tended to produce better solutions. However, the RCBP method is both faster and gives better solutions than the DBP approach, and in some cases than the DBONP method.

Note that we never employed a specialized branch-and-bound technique, except for that native to CPlex in the restricted master problem and subproblems. It therefore appears that the combination of DBP and O'Neill pricing approaches may facilitate the develop-
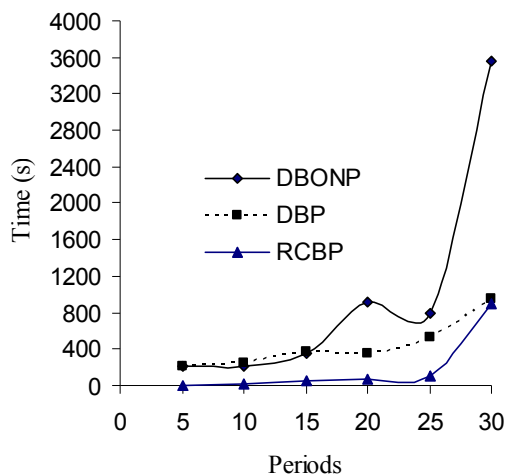
Figure 10: *Solution times required to solve DBP, DBONP, and RCBP.*

ment of a range of new column generation algorithms that may prove effective for integer programs.

References

[1] DE CARVALHO JMV, 1998, *Exact solution of cutting stock problems using column generation and branch-and-bound*, International Transactions in Operational Research, **5(1)**, pp. 35–44.

[2] FOURER R, GAY DM & KERNIGHAN BW, 1993, *AMPL: A modelling language for mathematical programming*, Curt Hinrichs Publishing, Pacific Grove (CA).

[3] GEOFFRION AM, 1974, *Lagrangean relaxation for integer programming*, Mathematical Programming Study, **2**, pp. 82–114.

[4] GOMORY RE & BAUMOL WJ, 1960, *Integer programming and pricing,* Econometrica, **28(3)**, pp. 521–550.

[5] GUNN EA, MILLAR HH & NEWBOLD SM, 1991, *A model for planning harvesting and marketing activities for integrated fishing firms under an enterprise allocation scheme*, European Journal of Operational Research, **55(2)**, pp. 243–259.

[6] HASAN MB AND RAFFENSPERGER JF, 2006, *A mixed integer linear program for an integrated fishery*, ORiON, **22(1)**, pp. 19–34.

[7] HASAN MB & RAFFENSPERGER JF, 2007, *A decomposition based pricing method for solving a large-scale MILP model for an integrated fishery*, Journal of Applied Mathematics and Decision Science, **2007**, Article ID 56404, Available from: `http://www.hindawi.com/journals/jamds/volume-2007/regular.1.html`.

[8] JENSSON P, 1988, *Daily production planning in fish processing firms*, European Journal of Operational Research, **36(3)**, pp. 410–415.

[9] MAMER JW & MCBRIDE RD, 2000, *A decomposition-based pricing procedure for large-scale linear programs: An application to the linear multi-commodity flow problem*, Management Science, **46(5)**, pp. 693–709.

[10] MARTIN K & SWEENEY DJ, 1983, *An ideal column algorithm for integer programs with special ordered sets of variables*, Mathematical Programming, **26(1)**, pp. 48–63.

[11] MARTIN KR, SWEENEY DJ & DOHERTY ME, 1985, *The reduced cost branch-and-bound algorithm for mixed integer programming*, Computers & Operations Research, **12(2)**, pp. 139–149.

[12] MIKALSEN B & VASSDAL T, 1981, *A short term production planning model in fish processing*, pp. 223–233, in HALEY KB (ED.), *Applied operations research in fishing*, Plenum Press, New York (NY).

[13] O'Neill RP, Sotkiewicz PM, Hobbs BF, Rothkopf MH & Stewart WR, 2005, *Efficient market-clearing prices in markets with non-convexities,* European Journal of Operational Research, **164(1)**, pp. 269–285.

[14] Raffensperger JF & Schrage L, 2008, *Scheduling training for a tank battalion: How to measure readiness*, Computers and Operations Research, **35(6)**, pp. 1844–1864.