

BAB II

LANDASAN TEORI

Penelitian ini dibuat dengan adanya teori pendukung agar sistem dapat berjalan sesuai dengan apa yang sudah direncanakan. Pada bab ini akan dibahas tentang teori penjadwalan dan perutean AGV (*Automated Guided Vehicles*) yang berhubungan dengan penjelasan sistem-sistem yang akan dipakai pada penelitian ini. Adapun teori-teori yang akan digunakan dalam penelitian ini sebagai berikut.

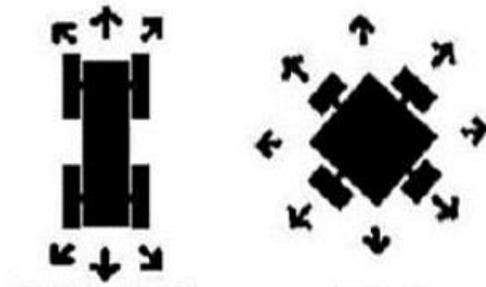
2.1 Automated Guided Vehicle (AGV)

Automated Guided Vehicles (AGV), juga dikenal sebagai Autonomous Mobile Robots (AMR), adalah jenis kendaraan otomatis yang dirancang untuk bergerak dan beroperasi secara mandiri tanpa campur tangan manusia. AGV ini biasanya digunakan dalam lingkungan industri dan logistik untuk mengotomatisasi proses internal transportasi barang dan material. Karakteristik utama dari AGV termasuk kemampuan untuk bergerak sendiri dengan menggunakan sistem navigasi seperti sensor, laser, dan kamera untuk mengenali dan mengikuti rute yang telah ditentukan. AGV berperan penting dalam revolusi otomatisasi industri. *Automated Guided Vehicles* (AGVs) menggunakan roda segala arah, panduan sistem AGV melalui jalur dalam navigasi. Roda *omni-directional* tipe mecanum dengan 4 buah pada AGV. Setiap roda dikendalikan oleh motor DC memungkinkan AGV untuk bergerak ke segala arah secara bersamaan mandiri. Sistem panduan jalur menggunakan Sensor garis yang membaca garis putih untuk memandu AGV dalam proses membawa alat stasiun ke stasiun lain. Fungsi Station adalah tempat menyimpan dan menyimpan barang Ambil kotak peralatan. Hasilnya menunjukkan Menggunakan 4 roda Mecanum dapat diterapkan AGV sedang melakukan penanganan alat, AGV Berkendara di sepanjang rute dengan kecepatan (0,5 – 1,8) m/s, model AGV dilihat pada **Gambar 2.1**[16].



Gambar 2.1 Model AGV

Jenis AGV yang paling banyak digunakan adalah Pengikut garis AGV, yaitu dari Ada jalan yang mengarah dari satu tempat ke tempat lain Terletak di sepanjang jalur AGV. Bimbingan jalan Bisa berupa kawat berwarna, strip magnet dan laser (Piyare & Singh) Model navigasi yang dipandu jalur ini Biasanya digunakan dalam sistem AGV sederhana. Navigasi dirancang untuk mode ini tidak bergerak.

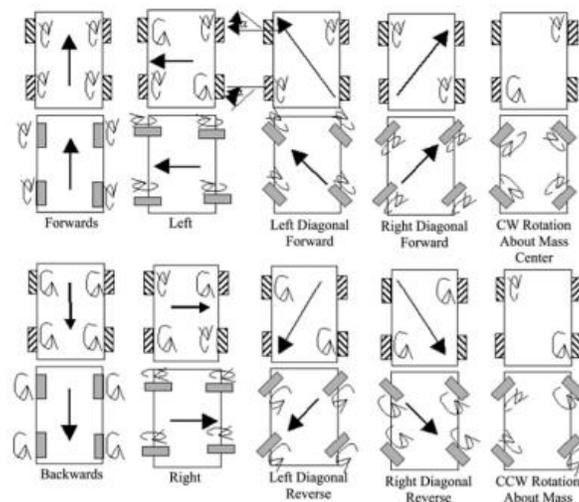


Gambar 2.2 Perbandingan arah gerak AGV dengan roda standar dan omni-directional

Secara umum, desain AGV menggunakan Roda standar, yaitu roda dengan gerakan nonholonomis. Dalam model seperti itu, gerak tidak Tingkat kebebasan apa pun dapat dikontrol bergerak secara mandiri, jadi hanya dapat bergerak ke arah tertentu dijadwalkan. Meskipun di umumnya dapat menjangkau setiap lokasi, dan orientasi spasial dua dimensi, tetapi membutuhkan manuver yang kompleks dan perencanaan jalan kompleks. Di samping itu, Perancangan sistem mengatasi hambatan adalah tugas yang menakutkan, penggunaan roda *omni-directional* memungkinkan semua bisa digerakkan secara mandiri di AGV orientasi mudah ditentukan tanpa reorientasi dapat dilihat pada **Gambar 2.2**[17].

2.1.1 Pengendalian gerak AGV menggunakan Mecanum Wheel

Roda mecanum dapat melakukan gerakan omni-directional dengan menggunakan penggerak motor DC yang bersifat lebih fleksibel dan kemampuan manuver yang baik daripada roda biasa. Roda mecanum memiliki potensi besar dalam beberapa situasi seperti ruangan yang kecil. Roda mecanum ini berbeda dari roda konvensional pada umumnya. Dalam hal tersebut roda mecanum memiliki rol kecil pada sekitar rodanya, yaitu 45° ke bidang roda dan 45° ke garis melalui pusat roller yang sejajar dengan sumbu rotasi roda. Dan memiliki ukuran diameter 6 cm dan lebar 3 cm, AGV memakai empat buah roda mecanum atau omniwheels yang masing-masing roda terhubung langsung ke motor DC dan motor driver untuk setiap 2 motor DC.

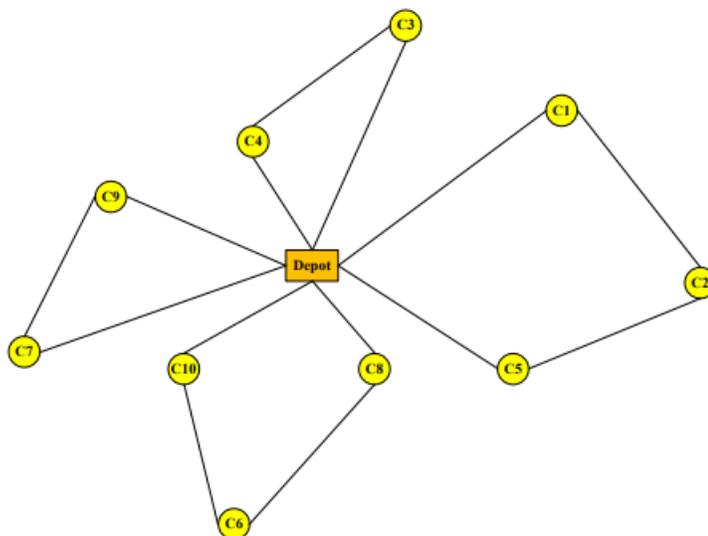


Gambar 2.3 Gerakan Mecanum Wheel

Motor DC yang dipakai memiliki beberapa keunggulan, termasuk kinerja awalan dan pengereman yang baik, putaran panjang, pengaturan kecepatan yang sangat baik sehingga mudah kontrol. Dengan menggunakan kombinasi dari empat buah roda ini, robot AGV dapat bergerak ke arah omni tergantung pada arah rotasi setiap roda. Menggerakkan semua roda ke arah yang sama menghasilkan gerakan maju ataupun mundur. Menjalankan dua roda diagonal searah jarum jam dan dua roda diagonal lainnya berlawanan arah jarum jam atau sebaliknya, akan menghasilkan gerakan ke kanan ataupun kekiri. Menjalankan dua roda pada sisi yang sama searah jarum jam dengan roda di sisi lainnya berlawanan arah jarum jam menyebabkan gerakan memutar. Kombinasi gerakan roda tersebut akan berfungsi untuk memaanuver gerakan robot AGV ke segala arah dengan putaran apa pun. **Gambar 2.3** menunjukkan perbandingan kombinasi rotasi roda untuk fungsi pergerakan antara roda mecanum dengan roda konvensional[18].

2.2 Scheduling

Scheduling atau Penjadwalan dalam AGV adalah sesuatu Proses perencanaan di mana kendaraan akan beroperasi. Dispatch mengeluarkan perintah ke kendaraan berdasarkan tujuan, mis. Meminimalkan biaya dengan memperhatikan kendala yang ada seperti permintaan pelanggan atau kapasitas kendaraan pola kordinat perjalanan AGV dilihat pada **Gambar 2.4**[19].



Gambar 2.4 Contoh gambar pola pergerakan AGV

Scheduling merupakan permasalahan mencari jarak terkecil untuk mengunjungi semua node tepat satu kali pada suatu graph dan kembali ke node mula dengan satu kendaraan. Kemudian muncul permasalahan scheduling yang memiliki lebih dari satu kendaraan seperti **Gambar 2.4** diharuskan memenuhi permintaan masing-masing node. Setiap kendaraan dari node depot mengunjungi beberapa node dan kembali lagi ke node depot yang disebut tour. Adapun kendala tambahan yang umum untuk digunakan. Kendala kapasitas, yaitu untuk setiap kendaraan memiliki batasan kapasitas sehingga jumlah atau berat barang yang dibawa suatu kendaraan tidak lebih dari kapasitas yang ada:

1. Kendala kunjungan minimal, yaitu pada setiap tour harus mengunjungi sejumlah node yang lebih dari atau sama dengan minimal node yang ditetapkan.
2. Kendala total waktu, yaitu total waktu dalam melakukan suatu tour tidak boleh melebihi batas yang ditentukan.
3. Time windows, yaitu setiap node memiliki jam buka atau tutup sehingga hanya bias dikunjungi oleh kendaraan pada saat-saat tertentu saja .
4. Precedence relation yaitu adanya kendala pada beberapa pasangan pelanggan i dan j sehingga pelanggan i harus dikunjungi terlebih dahulu daripada pelanggan j dalam satu tour.

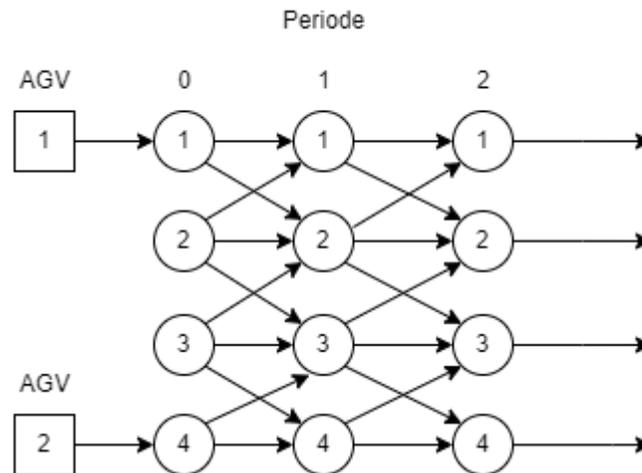
2.3 Routing

Perutean adalah perpanjangan dari pengiriman, tetapi dengan Bedanya ada node atau simpul yang tidak perlu dikunjungi Atau memiliki permintaan. Node yang perlu dikunjungi dan diimplementasikan Permintaan masuk sebagai gudang atau klien, sedangkan node tidak perlu dikunjungi sebagai persimpangan, bundaran atau belokan kembali. Pertanyaan ini lebih untuk implementasi nyata. Masalah perutean juga termasuk dalam taksonomi Masalah kompleksitas waktu komputasi di NP. Namun Masalah ini dapat dibagi menjadi dua masalah, yaitu penjadwalan Dan cari jarak terpendek antara pelanggan dan gudang. Sehingga hitung jarak terpendek antara

mencari masing-masing Node pos pertama dan node pos selanjutnya, lalu akan berubah sesuai jadwal[20].

2.4 Strategi Pencarian

Branch-and-Bound adalah pendekatan umum yang digunakan untuk menyelesaikan masalah optimisasi atau pencarian ruang solusi dengan cara sistematis. Ide dasarnya adalah membagi masalah menjadi bagian-bagian yang lebih kecil, mengevaluasi nilai atau kualitas potensial dari setiap bagian tersebut, dan menggunakan informasi ini untuk memutuskan apakah harus melanjutkan pencarian di bagian-bagian tertentu atau mengabaikannya.[21].



Gambar 2.5 Jaring Ruang Dan Waktu

Berikut adalah langkah-langkah umum yang terlibat dalam strategi pencarian Branch-and-Bound:

1. Pembagian (Branching): Tahap awal adalah membagi masalah awal menjadi submasalah lebih kecil yang bisa dianalisis dengan lebih baik. Ini biasanya dilakukan dengan memilih satu variabel atau komponen dari masalah dan membaginya menjadi beberapa bagian.

2. Penilaian (Bounding): Pada setiap cabang atau submasalah, tahap ini melibatkan penilaian potensi solusi optimal yang dapat dicapai di dalam submasalah tersebut. Ini dilakukan dengan menghitung batasan atas (upper bounds) dan batasan bawah (lower bounds) pada nilai solusi di submasalah tersebut. Upper bound adalah nilai maksimum yang dapat dicapai dalam submasalah, sementara lower bound adalah perkiraan terbaik untuk nilai solusi optimal.
3. Pemotongan (Pruning): Pada tahap ini, cabang-cabang yang tidak berpotensi menghasilkan solusi yang lebih baik dari solusi terbaik yang sudah ditemukan selama pencarian dapat diabaikan. Ini membantu mengurangi jumlah evaluasi yang perlu dilakukan.
4. Penyebaran (Exploration): Pencarian dilanjutkan pada cabang-cabang yang masih memiliki potensi untuk menghasilkan solusi yang lebih baik daripada solusi terbaik yang telah ditemukan. Langkah ini melibatkan ekspansi cabang-cabang ini menjadi submasalah lebih lanjut dan dilanjutkan dengan langkah-langkah sebelumnya.
5. Berhenti (Termination): Pencarian berlanjut hingga salah satu kondisi berhenti tercapai. Kondisi ini bisa berupa mencapai solusi optimal yang memenuhi kriteria yang ditetapkan atau ketika seluruh pohon pencarian telah dijelajahi dan tidak ada lagi cabang yang harus dieksplorasi.

Strategi ini sangat bergantung pada pemilihan cabang yang tepat untuk dieksplorasi lebih lanjut dan penggunaan informasi batasan atas dan batasan bawah untuk menghindari evaluasi yang tidak perlu. Penggunaan batasan waktu juga dapat dimasukkan untuk membatasi waktu yang dihabiskan pada setiap submasalah. Secara umum, metode Branch-and-Bound sangat fleksibel dan dapat disesuaikan dengan berbagai jenis masalah optimisasi atau pencarian, dari permasalahan penjadwalan hingga pemecahan persamaan.

2.5 Metode Constraint Programing

Menentukan kendaraan mana yang akan memproses permintaan penanganan material apa pada waktu tertentu dengan menghasilkan tugas tugas yang dipesan ke AGV sambil meminimalkan jumlah total penundaan pengiriman. Jumlah total keterlambatan diukur dengan menjumlahkan perbedaan antara waktu mulai yang direncanakan dan waktu mulai paling awal dari setiap pengiriman. Pada model ini matriks jarak (waktu) diperoleh dengan menggunakan jalur terpendek antar node. Dengan demikian, penundaan yang dihitung (yang tidak memperhitungkan kemungkinan konflik) memberikan batas bawah penundaan yang sebenarnya.[22].

Oleh karena itu, posisi awal (awal cakrawala) dan akhir (akhir cakrawala) dapat dilihat sebagai simpul tugas tiruan untuk setiap AGV untuk cakrawala saat ini dan yang berikutnya. Mendefinisikan set W sebagai set semua tugas termasuk yang mulai dummy. Set dan Parameter Model CP dimodelkan sebagai berikut $W =$ Set Untuk tugas penjemputan dan Pengiriman, $W^p =$ Set Tugas penjemputan, $W^d =$ Set Tugas pengiriman, $V =$ Set Untuk intruksi AGV, $R =$ Set permintaan : masing-masing adalah sepasang penjemputan (r^p) dan pengiriman (r^d), $T =$ Himpunan periode waktu, $I =$ set berisi R dummy start meminta, $O =$ set berisi R dan dummy end meminta, $P =$ Himpunan pasangan tugas yang dihubungkan oleh hubungan prioritas (p^1, p^2), $E =$ Himpunan semua waktu mulai paling awal, $D =$ Panjang jalur terpendek antara simpul dari dua tugas. Variable yang digunakan $vr \in V$ variabel mewakili AGV yang ditugaskan untuk meminta r , $sr \in O$ variabel mewakili permintaan yang dilakukan segera setelah r pada kendaraan yang sama, $tr \in T$ variabel mewakili waktu mulai permintaan r . Fungsi tujuannya yaitu terdiri dari meminimalkan jumlah perbedaan antara waktu mulai awal yang diberikan dan waktu mulai sebenarnya dari pengiriman untuk model cp sebagai berikut:

$$\text{Min } \sum_{j \in W^d} (t_j - E_j) \quad (1)$$

$$\text{s.t. } v_o = v_{s_o} \quad \forall o \in O, \quad (2)$$

$$t_{r^p} + 1 + D(r^p, r^d) \leq t_{r^d} \quad \forall r \in R, \quad (3)$$

$$s_{r_1} = r_2 \Rightarrow t_{r_1^d} + 1 + D(r_1^d, r_2^p) \leq t_{r_2^p} \quad \forall r_1, r_2 \in R, \quad (4)$$

$$\neg(t_{p^1} \leq t_i \leq t_{p^2}) \quad \forall p \in P, i \in W : (p^1, p^2 \neq i \wedge (n_i = n_{p^1} = n_{p^2})), \quad (5)$$

$$t_{p^1} + 1 \leq t_{p^2} \quad \forall p \in P : (p^1 \in W^p) \wedge (p^2 \in W^d), \quad (6)$$

$$t_{p^1} + 1 + e_{p^1} \leq t_{p^2} \quad \forall p \in P : (p^1 \in W^d) \wedge (p^2 \in W^p), \quad (7)$$

$$(t_i \geq t_j + 1) \vee (t_i + 1 \leq t_j) \quad \forall i, j \in W : (i \neq j) \wedge (n_i = n_j). \quad (8)$$

Langkah langkah utama dari algoritma constraint programming untuk penjadwalan di tunjukan pada persamaan (2) untuk memastikan bahwa setiap permintaan ditugaskan ke AGV yang sama. (3) Menetapkan untuk setiap AGV yang sedang memproses permintaan harus memiliki cukup waktu untuk pergi dari node penjemputan permintaan ke node pengiriman. (4) Memastikan, jika satu permintaan adalah lanjutan proses dari permintaan lain pada AGV yang sama. (5) Jika dua tugas yang dihubungkan oleh batasan prioritas berbagi simpul yang sama, harus diproses secara berurutan pada simpul. (6) Menyatakan setidaknya satu periode (durasi pengambilan) harus berlalu antara waktu mulai pengambilan dan tugas pengiriman yang dihubungkan oleh batasan prioritas. Selain itu, tidak ada tugas lain yang dapat dilakukan pada sebuah node antara pelaksanaan dua tugas yang dihubungkan oleh kendala prioritas. (7) Memastikan, jika pengiriman mendahului pengambilan, setidaknya ada satu periode (durasi pengiriman) ditambah waktu pemrosesan yang dikirim antara waktu mulainya. (8) Memastikan bahwa dua tugas yang berbeda tidak dapat dimulai pada node yang sama di waktu bersamaan.

Permintaan transportasi terdiri dari penjemputan dan tugas pengiriman. Untuk tujuan pemodelan, permintaan awal dan akhir (dan tugas) dikaitkan dengan setiap AGV. Untuk membantu model CP, Menerapkan beberapa heuristik seleksi yang digunakan dalam kombinasi dengan pencarian branch and bound.

2.6 Metode Mix Integer Programming

Karena AGV dapat mengunjungi busur atau simpul lebih dari satu kali selama cakrawala, grafik ruang- waktu digunakan untuk memodelkan perutean AGV. Grafik ruang-waktu diilustrasikan pada **Gambar 2.5** dan dapat dijelaskan sebagai berikut: sebuah node didefinisikan untuk setiap periode (satu unit waktu) dan setiap titik akhir dari segmen jalur panduan. Setiap node dari periode tertentu dihubungkan oleh busur ke node berdekatan yang sesuai dari periode berikutnya. Busur tersebut sesuai dengan pergerakan AGV antara dua titik akhir yang berdekatan dari segmen jalur panduan. Setiap simpul pada periode tertentu juga dihubungkan oleh busur ke simpul yang sama yang bersesuaian pada periode berikutnya (yaitu busur horizontal). Ini sesuai dengan menunggu satu unit waktu di simpul itu. Tidak ada busur antara node dari periode yang sama. Semua busur grafik memiliki panjang satu satuan waktu. Untuk setiap AGV, ada simpul awal pada periode 0. Set dan parameter model MIP, V set AGV, N set node, T set $A+$ himpunan semua arc (termasuk waiting arc), f himpunan arc yang datang dari node I , $At[.]$ himpunan arc memasuki simpul I , $Aw[.]$ busur tunggu yang terkait dengan setiap simpul, $Ao[.]$ busur yang berlawanan dari setiap busur, n simpul awal AGV v , W larik catatan yang menjelaskan setiap tugas dengan dua bidang (simpul, waktu paling awal) v, t data diperoleh dari model CP, R himpunan permintaan: each adalah pasangan pick-up (rp) dan delivery (rd), $R+[., t]$ untuk setiap pasangan (node i , periode t), himpunan tugas penjemputan dan pengiriman r yang simpul layanannya adalah i dan yang dilakukan pada waktu t ($(t = tr - 1) \wedge (i = nr)$).[23] Variabel Mix Integer Programming yang di gunakan $X_{t,k,a}$ Variabel Boolean=1 jika AGV k mulai melintasi busur a pada periode t dan 0 atau sebaliknya. Karena yang di cari pada metode MIP ini hanya mencari solusi yang layak, menggunakan fungsi tujuan konstan. Modelnya adalah sebagai berikut:

$$\text{Min } 1 \quad (9)$$

$$\text{s.t. } \sum_{a \in A^f[n^k]} X_{k,a}^o = 1 \quad \forall k \in V, \quad (10)$$

$$\sum_{a \in A^+} X_{k,a}^t = 1 \quad \forall t \in T, k \in V, \quad (11)$$

$$\sum_{a \in A^f[i]} X_{k,a}^t + \sum_{a \in A^t[i]} X_{k,a}^t = 0 \quad \forall i \in N, k \in V, t \in [1 \dots (M-1)], \quad (12)$$

$$\sum_{k \in V} X_{k,a}^t + \sum_{k \in V} X_{k,A^o[a]}^t \leq 1 \quad \forall t \in T, a \in A, \quad (13)$$

$$X_{V_r, A^w[n_r, p]}^{T_r, p} = 1 \quad \forall r \in R, \quad (14)$$

$$X_{V_r, A^w[n_r, d]}^{T_r, d} = 1 \quad \forall r \in R, \quad (15)$$

$$\sum_{k \in V, a \in A^t[i]} X_{k,a}^t \leq 1 + \left(\sum_{r \in R^+[t-1, i]} 1 \right) \times \left(\sum_{r \in R^+[t, i]} 1 \right) \quad \forall i \in N, t \in T^+. \quad (16)$$

Langkah langkah utama dari algoritma Mix Integer Programming untuk perutean AGV di tunjukan pada persamaan (10) memastikan bahwa setiap AGV terletak pada posisi awalnya di awal horizon. Sementara persamaan(11) memastikan bahwa setiap AGV terletak pada posisi yang berbeda pada setiap periode. Persamaan (12) merupakan kendala konservasi aliran. (13) Menjadi kendala untuk penghindaran tabrakan. Persamaan (14) menyatakan bahwa setiap tugas penjemputan harus dilakukan oleh kendaraan yang tepat pada waktu yang tepat sedangkan, Persamaan (15) setara dengan kendala (14) untuk tugas pengiriman. Persamaan (16) menjadi kendala kapasitas simpul yang menyatakan bahwa hanya ada satu AGV pada sebuah simpul setiap saat kecuali kasus di mana satu AGV meninggalkan tugas simpul sementara yang lain baru saja masuk dalam tugas simpul yang sama. Jika MIP sudah menghasilkan solusi yang layak maka optimalitas untuk masalah global diperoleh. Jika tidak, pemotongan ditambahkan ke masalah utama yang diselesaikan kembali.

2.7 Pendekatan hibrid CP/MIP

Hooker dkk[24]. Menyajikan ulasan komprehensif tentang metode hibrid. Kami fokus di sini pada karya yang lebih relevan. Hooker memberikan wawasan tentang bagaimana CP dapat diintegrasikan ke dekomposisi Benders. Sejumlah peneliti telah mengintegrasikan CP dalam dekomposisi Benders klasik untuk MIP. Eremin

dkk[25]. Juga menyajikan metode dekomposisi hibrid di mana masalah utama diselesaikan dengan CP. Kepentingan utama dari pendekatan mereka adalah bahwa pengguna hanya perlu menentukan variabel dari setiap sub masalah.

Hal ini memungkinkan derivasi otomatis dari bentuk ganda dari setiap sub masalah dan ekstraksi otomatis dari dekomposisi Benders. Ini dapat membantu peneliti fokus pada CP atau pemrograman matematika (bukan kedua bidang) untuk merancang prototipe model dengan cepat. Pengembangan pendekatan berasal dari keterbatasan pendekatan pemrograman matematis, dasar untuk dekomposisi CP/MIP ini adalah memanfaatkan kekuatan CP untuk penjadwalan (karena dapat menangani kendala nonlinear) dan MIP untuk perutean dalam konteks khusus ini. Pendekatan ini terinspirasi oleh dekomposisi Benders berbasis logika meskipun tidak memanfaatkan dualitas[26].

Kendala yang ditambahkan ke model untuk menghilangkan solusi CP yang tidak memiliki solusi perutean yang layak (baik solusi CP yang ada dan sebanyak mungkin solusi CP lainnya dengan nilai tujuan yang sama (penundaan total). Satu pemotongan terdiri dari disjungsi tiga opsi: dengan dua opsi pertama, penugasan permintaan yang sama ke AGV disimpan tetapi waktu transportasi atau waktu mulai dari setidaknya satu pengiriman bertambah satu unit, opsi ketiga terdiri dari membuat perubahan pada penugasan permintaan ke AGV (setidaknya satu permintaan harus dipindahkan)[27].

Penguraian CP/MIP ini adalah untuk manfaat dari kekuatan CP untuk penjadwalan (karena dapat menangani kendala nonlinier) dan MIP untuk perutean dalam konteks khusus ini. Pendekatan ini terinspirasi oleh dekomposisi Benders berbasis logika. Dalam metode dekomposisi yang dikembangkan di sini, masalah utama CP menentukan penugasan keduanya permintaan transportasi ke kendaraan dan jadwal (yaitu waktu yang diharapkan) penjemputan dan pengiriman berdasarkan rute jalur terpendek (mengabaikan kemungkinan konflik rute). Untuk setiap solusi. Dari masalah master, sub masalah MIP mencoba menemukan rute bebas tabrakan yang memenuhi jadwal diperoleh dari masalah utama. Ketika tidak ada solusi (yaitu, tidak

mungkin menemukan solusi yang bebas konflik rute yang memenuhi jadwal), pemotongan logika dihasilkan dan dikirim kembali ke masalah utama[28].